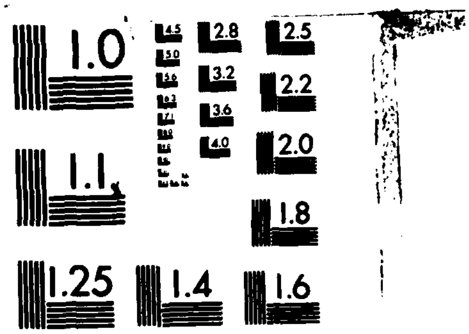
AD-A140 299   ADVANCED MOTOR-CONTROLLER DEVELOPMENT(U) WESTINGHOUSE   1/4
OCEANIC DIV ANNAPOLIS MD L E LESSTER ET AL. 22 JUN 83
DTNSRDC-PASD-CR-1-83 N00167-82-C-0122

UNCLASSIFIED   F/G 9/3   NL



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

(1)

AD A140299

ADVANCED MOTOR-CONTROLLER DEVELOPMENT

Final Report for Period October 1979 - June 1983

June 22, 1983

Report DTNSRDC-PASD-CR-1-83
Contract N00167-82-C-0122

L.E. Lesster
D.B. Zeitlin
W.B. Hall

Prepared for:

David W. Taylor Ship Research and Development Center
Annapolis Laboratory
Annapolis, Maryland 21402

by:

WESTINGHOUSE ELECTRIC CORPORATION
Oceanic Division
P.O. Box 1488
Annapolis, Maryland 21404

Table of Contents

This volume is divided into three separate reports which document the three stages of development.

Volume Summary

The three reports assembled in this volume represent work performed periodically over a span of 5 years, as part of a step-by-step development program.

The purpose of ~~the~~ H.'s development program was to investigate a promising alternative technique for control of a squirrel cage induction motor for subsea propulsion or hydraulic power applications. The technique uses microprocessor based generation of the pulse width modulation waveforms, which in turn permits use of a true integral volt-second pulse width control for the generation of low harmonic content sine waves from a 3 phase "Graetz" transistor power bridge.

In conjunction with the PWM Technique the novel use of a microprocessor based calculation of a signal to represent air gap flux in the motor was attempted. Externally this involves only sampling output currents and voltages to the motor and does not involve intrusion into the motor to place sense windings or Hall effect devices.

The purpose of this latter investigation was to generate a signal proportional to air gap flux which could be used in feedback control of the PWM microprocessor to vary the output pulsewidths of the bridge. This in turn could be used to maintain the air gap flux in the motor constant despite varying load.

During development it became apparent that keeping the air gap flux level constant did not represent an efficient control mechanism for the motor. This technique evolved from the normal use of induction motors on fixed frequency where better performance could be obtained on heavy load if the supply voltage could be boosted to maintain air gap flux despite increased voltage drop in the stator resistance and leakage inductance.

In reality an unloaded or lightly loaded induction motor can be run at very much lower voltages than 'nominal' for any given frequency and the purpose of feedback should be to increase the voltage approximately with load to maintain just sufficient flux to prevent under-excited operation with consequent high slip frequencies.

The use of the air gap flux synthesis technique appeared to work in principle, but difficulties were encountered in using it at low frequencies. Calculation of the feedback signal was very sensitive to the phase relationship of the motor current and voltage, and distortion in the feedback pre-sampling filters which is evident in the low frequency oscilloscope pictures (Phase II, Task B Report) is of serious consequence here. Further, the air gap flux signal as derived via a reactive power calculation is double valued and for use as a regulation signal, care must be taken to ensure that the load remains always on the over-exited side or always on the under-exited side of the optimum excitation level for any particular load and frequency. While this can be assumed statically, in a dynamic loading situation, unless the feedback loop is fast compared with load change this cannot be assured.

For purposes of checking out other key points of the motor controller development independently of the flux feedback signal, a slight modification of the feedback signal was made. This permitted sensing and feedback of the motor load current and provided satisfactory demonstration and of overload protection by ramping down in frequency. During this mode of operation the output voltage was controlled to have essentially constant volts/Hz except for the drop due to stator resistive and reactive components. Satisfactory operation in this mode indicates that regulation at some constant air gap flux would be possible although not necessarily desireable from an optimum control point of view.

In summary, the following was achieved:

1. Full microprocessor based, sine wave, 3 phase, variable frequency, variable amplitude output waveform generation with volt-second control of the pulsewidths based on $\int vdt$ calculations. This was performed using an 8 bit M6800 processor in conjunction with pulse timer modules in an interrupt driven configuration.

2. Proof of principle of air gap flux synthesis technique and demonstration of overload ramp down dynamic load shedding.

3. Construction and test of a 7.5 kw 3 phase transistor bridge driving a 32 Hz induction motor.

4. Test of the complete motor controller on a dynamometer, and test of all principles involved.

These tests showed a sound hardware design which provided high quality sine wave currents to the motor over extended periods of time at all power levels. These tests are showed that with some extra investigation in the feedback area, an extremely viable, high efficiency motor controller for deep ocean application has been developed.

ADVANCED MOTOR CONTROLLER

PHASE I REPORT

TABLE OF CONTENTS

MICROPROCESSOR - BASED THREE PHASE MOTOR CONTROLLER

Introduction

The objective of this study was to define an approach to digital synthesis of waveforms for a low harmonic content sine wave output PWM 3Ø variable frequency transistorized induction motor drive inverter, to analyze an air gap flux synthesis system using microprocessor supervisory control and to provide block diagram information and block response characteristics for computer modeling of the motor drive system.

Summary & Recommendation

Initially it was thought that an M6800 microprocessor would be too slow for PWM generation for this application. However, careful analysis of the capabilities of the processor as calculator of numbers rather than a generator of timing waveforms led to consideration of its use to calculate pulsewidths and timing intervals, numbers which can be used to control a single chip 3-channel timer. This component turns the numbers into serial outputs as a function of time - the pulse width modulation waveforms - which are amplified by the slave power switches of a 3Ø power bridge. Amplitude scaling of the output as a function of demanded motor voltage, controlled via the air gap flux feedback loop, is effected by digital multiplication using a high speed hardware multiplier.

The air gap flux synthesis and feedback control loop relies on sensing the voltages and currents to the motor and simulating the magnetizing component of the reactive power drawn by the motor. Since the output waveforms from the bridge to the motor are not smooth sinusoides, the sensed inputs from these must be filtered by equal filters to maintain phase relationship between current and voltage. Once filtered, analog/digital conversion turns these inputs into a form suitable for a microprocessor calculation routine which modifies the voltage output of the PWM drive in response to motor load requirements.

The PWM subsystem block was designed in software and hardware using an AMI 6800 microprocessor development system for the software development and an AMI prototyping general purpose microprocessor board for hardware. Pictures of waveforms are provided.

Imperfections in the triangle crossing modulation approach, which is an easy-to-implement technique in analog circuitry, were observed. The microprocessor approach to PWM permits use of an optimum integral technique which noticeably reduces harmonics.

The work performed in this study has covered the basic requirements for microprocessor control of a sine wave output motor drive system and in addition has developed a novel approach to PWM synthesis. It is recommended that further work in this area be addressed to tying the PWM control and air gap flux synthesis loops together in both software and hardware and reduction to practice of a full inverter/motor system.

Discussion

Flux Synthesis Control Loop

In attempting to control the air gap flux in an induction motor it is necessary to maintain a given magnetizing current in the magnetizing inductance component of the primary (stator) winding of the motor. In the equivalent circuit of the induction motor (Figure 1) it is the current I_m which is to be kept constant in the presence of varying load currents and hence varying voltage drops across R_1, L_1 and changes with temperature of winding resistances.

The input power to the motor can be considered to have two components. The true power VI cos $\emptyset$ and the reactive power VI sin $\emptyset$ or, if $V = V_1$ Sinwt & $I = I_1$ Sin(wt-$\emptyset$) the power components are 1/2 VI Cos$\emptyset$ and 1/2 VI Sin$\emptyset$. The true power input includes power lost in winding resistances and iron loss as well as in equivalent load resistor. The reactive power component, however, by definition is independent of all the resistance components and represents the product of the terminal voltage and the quadrature currents flowing in the leakage inductance L_1, the magnetizing inductance L_m and the secondary (rotor) inductance L_2.

These inductances are fixed once the motor is made and do not change with load, etc. It remains then to find the 'voltage drop' in L_1 due to I_m and I_2 so that the 'voltage' applied to L_m and hence the magnetizing current I_m can be found.

The flux control loop block diagram is shown in Figure 2, and the full derivation as well as the microprocessor assembly language program for flux loop control is given in Appendix A. During the development of the block diagram, numerical constants were selected on the basis of measurements made on the motor chosen for testing. Closed loop operation with that motor has not been performed and it is anticipated that some adjustments to the constants will be required for proper operation in that mode.

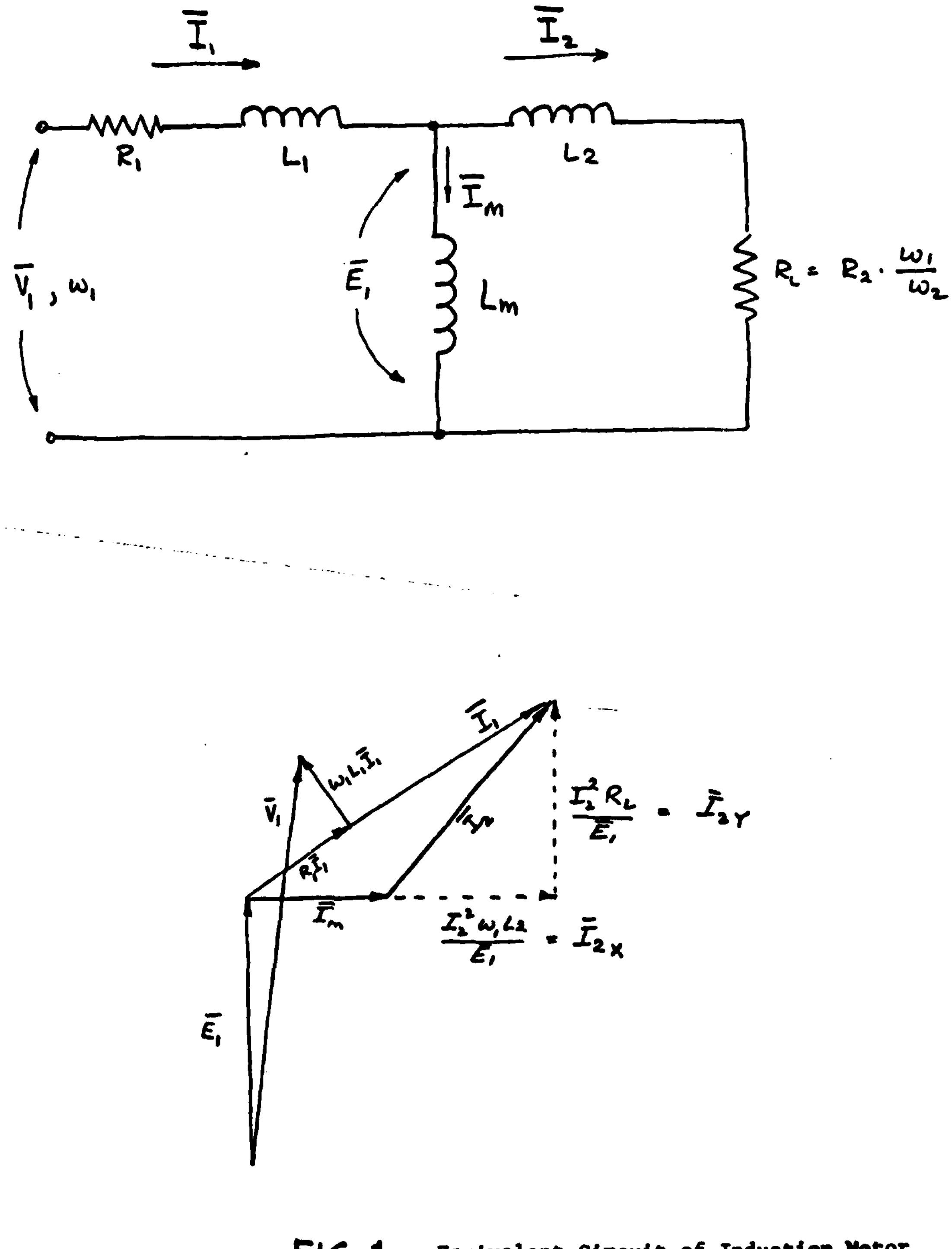


FIG 1 - **Equivalent Circuit of Induction Motor**

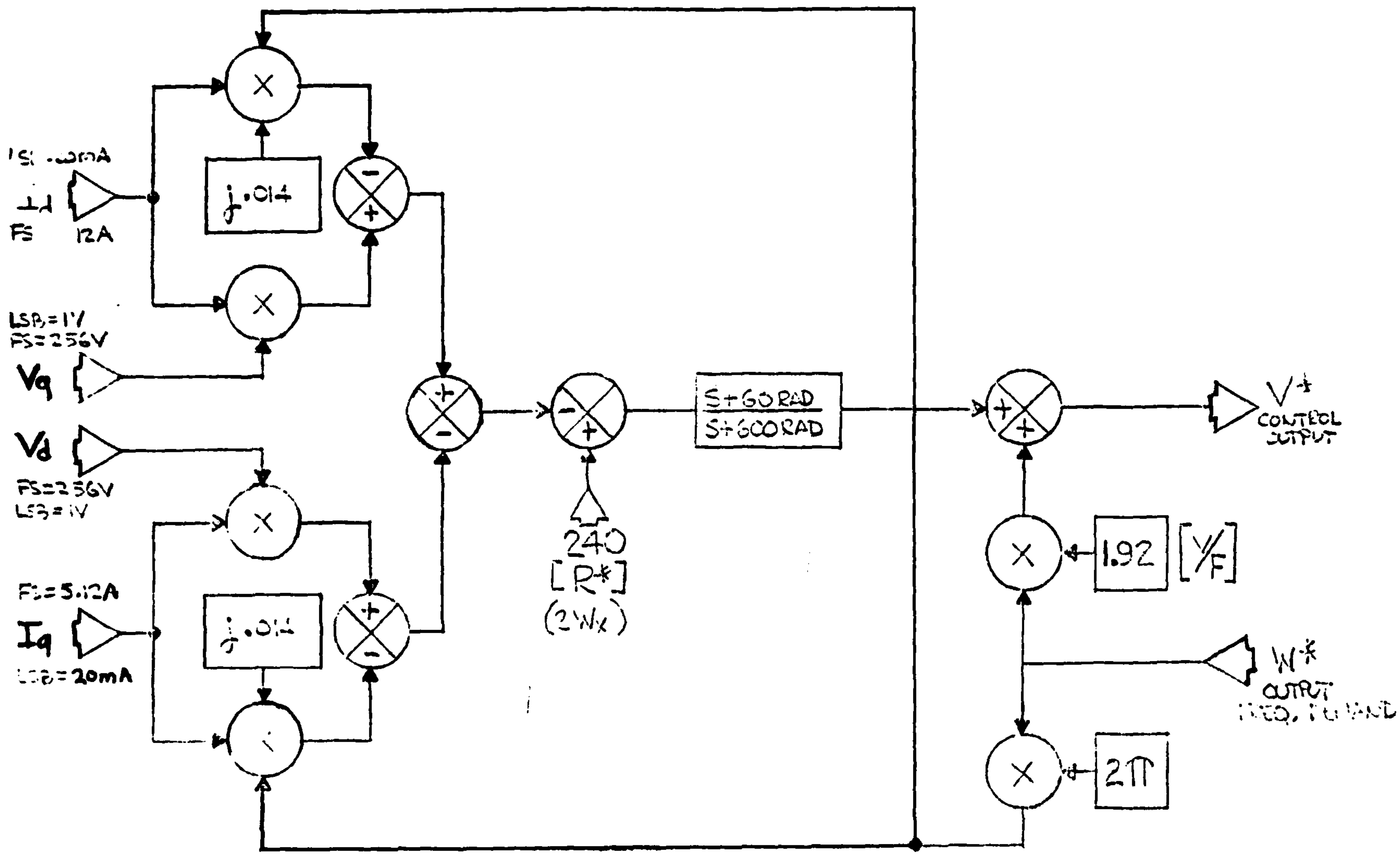


FIGURE 2
μP CONTROLLED AC MOTOR DRIVE
FLUX LOOP CALCULATION BLOCK DIAGRAM

Pulse Width Modulation Synthesis

Background

Initially it was thought that the M6800 would be too slow to generate the pulse width modulated wave forms. However, a closer look at the problem, as outlined in Appendix B, indicated that by using table look up and a hardware multiplier it should be possible to implement algorithms which generate time related bytes from which the serial PWM signals for the three phases can be generated. Such a system was devised and is described below.

Hardware

Figure 3 shows a functional block diagram of the hardware used to create the PWM waveforms. The central processing unit is an AMI 6800 with a 1 MHz clock, 8K bytes of EPROM for program and timing sequence data storage, 1K bytes of RAM for scratchpad storage, a Pulse Timer Module containing 3 independent timers with output signals, an 8x8 bit multiplier to speed up calculations and a keyboard interface used to command different voltages and frequencies.

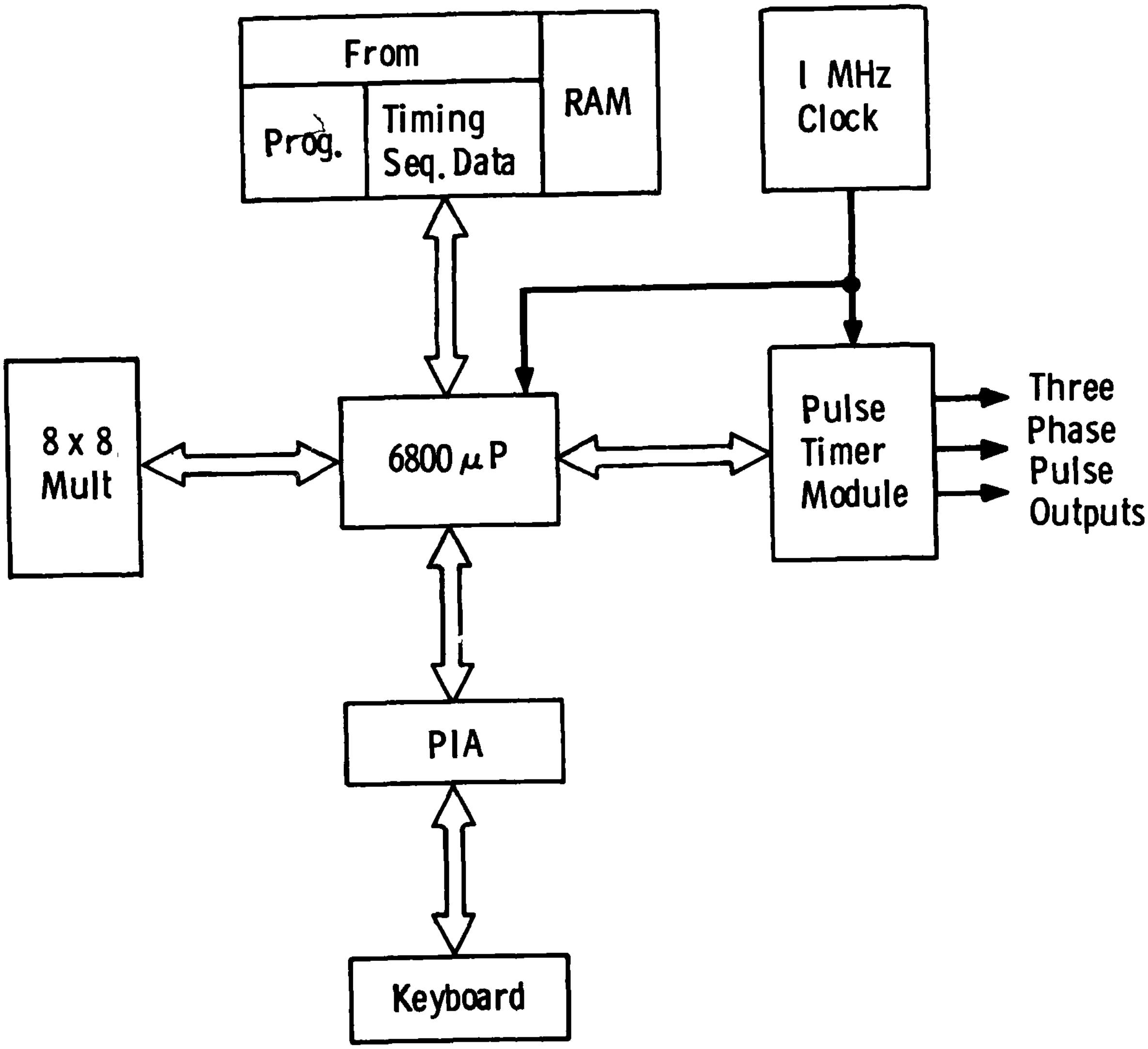


Figure 3 HARDWARE

Software

A frequency table is stored in PROM for each frequency that is to be output. It contains all the information required to generate the PWM waveforms for a number of voltage steps. Stored in the Frequency Table are:

1. The number of pulse periods in a 60 degree segment of the output sine wave (n).
2. The number of timer clocks in a pulse period (NCLKS).
3. A scale factor table containing the scaling data for a number of different voltage steps.
4. A baseline timing sequence for a given voltage containing, for each pulse period, the width of the dominant pulse (t_d), the width of the first complementary pulse (t_{ci}) and the centerpoint of the dominant pulse (CP).

These parameters are sufficient to generate timing sequences for a large number of voltage steps for a given frequency while at the same time keeping the memory requirements low.

Synthesizing the PWM waveforms for a given frequency and voltage requires generating the proper clock counts to the three timers in the Pulse Timer Module. (The timer outputs change state each time the counter reaches zero.) In order to do this, some temporary values must be defined. See Figure 4 for timing relationships.

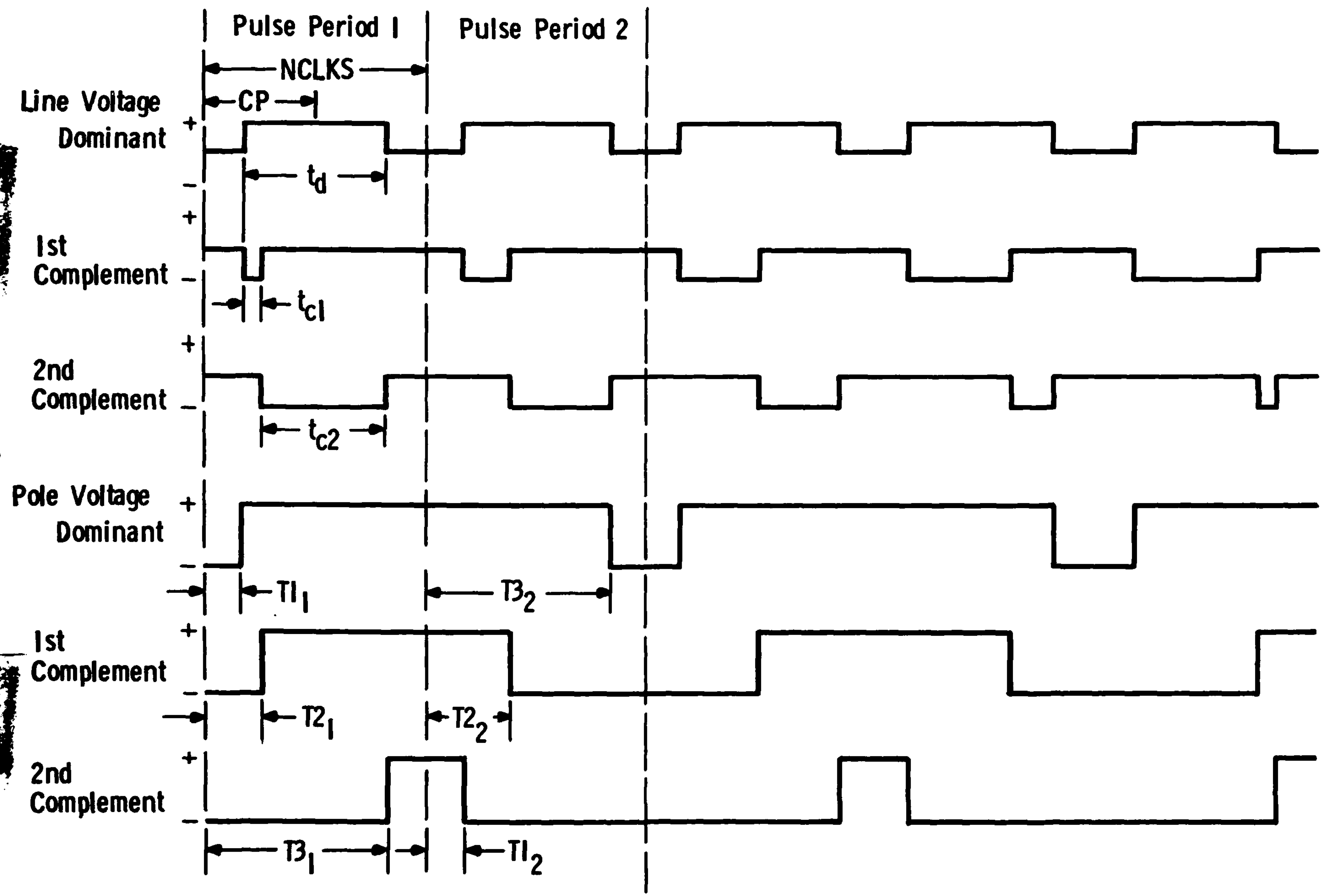


Figure 4 CONTROLLER TIMING DIAGRAM

T1 = time from start of a pulse period to the beginning of the dominant pulse.

T2 = time from start of a pulse period to the beginning of the 1st complementary pulse.

T3 = time from start of a pulse period to the beginning of the 2nd complementary pulse.

It can be seen that these times correspond to switching times where:

T2 = switching time of the 1st complementary pole voltage

T1 and T3 = switching time of the dominant or 2nd complementary pole depending on the pulse period.

The pulse lengths will vary proportionally with the amplitude of the output sine wave during the pulse interval. Therefore, the stored pulse widths in the baseline sequence must be scaled to the proper voltage. The scale factors are stored in the scale factor table contained in the frequency table and are computed such that:

$$SF = \frac{V\ desired}{V\ baseline}$$

The scaled pulsewidths now become:

$$t_d' = t_d * SF$$
$$t_{c1}' = t_{c1} * SF$$

where t_d and t_{c1} are the pulse widths stored in the baseline timing sequence for this pulse interval.

It can be seen that:

$$T1 = CP - t_d'/2$$

$$T2 = T1 + t_{c1}'$$

$$T3 = T1 + t_d'$$

In order to resolve the phase polarities and keep the optimal switching sequence, for the i^{th} pulse period,

$$TC1_i = T2$$

and, if i is odd

$$TD_i = T1_i$$

$$TC2_i = T3_i$$

else, if i is even

$$TD_i = T3_i$$

$$TC2_i = T1_i$$

where TD_i, $TC1_i$ and $TC2_i$ are the pole switching times of the dominant, 1st complementary and 2nd complementary phases, respectively.

Now, since a single timer controls each pole, the time difference between firings on each pole must be computed and saved in memory for the pulse timer module controller.

$$TD_i = TD_i - TD_{i-1} + NCLKS$$

$$TC1_i = TC1_i - TC1_{i-1} + NCLKS$$

$$TC2_i = TC2_i - TC2_{i-1} + NCLKS$$

These counts are computed for each pulse interval and saved in three memory buffers, one for each timer. Also saved is the time from the last switching of each phase to the end of the pulse interval ($NCLKS-TD_n$, $NCLKS-TC1_n$, $NCLKS-TC2_n$). This is done to keep the phase relationships constant by forcing each phase to execute an identical number of clocks in a 60° segment. Figure 5 shows the contents of the computed timing buffers.

With the timing counts now computed and stored in memory it becomes a matter of getting the proper count to the proper timer to preserve the phase relationships between the line voltages. Each of the three timers controls a single phase of the three phase signal.

When one of the timers counts down to zero it sends an interrupt to the processor. At that time the processor outputs the next count from the timing buffer to the timer latches, checks to see if this is the end of a 60° segment and sets up for the next interrupt. This procedure is followed by each of the three timers until the end of the 60° segment.

At the boundary between segments, the processor must resolve phases and insure that the timer clock counts for each of the three timers over the segment are identical. It does this by adding the remaining time in the last pulse interval (e.g. $NCLKS-TD_n$) to the first switching time of the proper phase (e.g. $TC2_1$) of the new timing buffer (see Figure 3). Also at this time, the phase identifier for the timer is updated to reflect its new status.

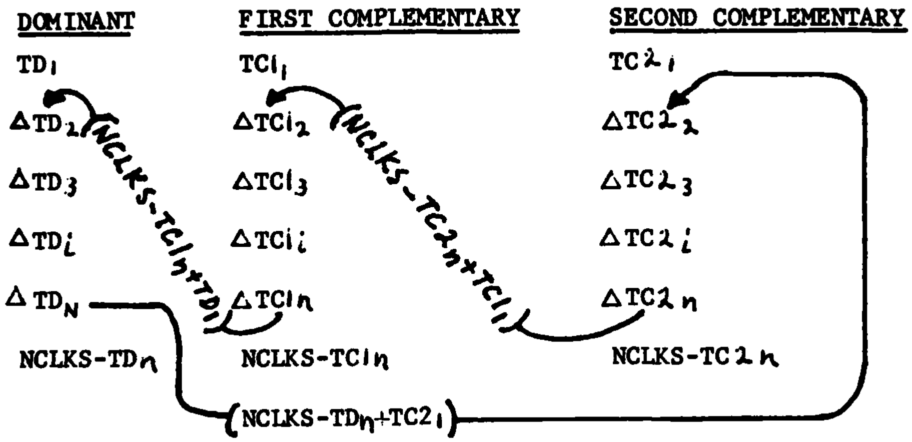


FIGURE 5. TIMING BUFFER CONTENTS AND SIGNAL FLOW FOR EACH OF THE THREE TIMERS

The boundary between segments is the only place that a new timing sequence can be started. This maintains the integrity of the system as well as insures the proper phasing relationships for all signals. Thus, if the same frequency were maintained over a 180° interval 3*NCLKS counts would be output to each timer.

Two separate and independent routines control the synthesis of the PWM waveforms. The Computer Timing Sequence (CTS) routine accepts as inputs the voltage and frequency commands, computes the counts required for the timer controller and saves the counts in a timing buffer. The Pulse Timer Module Interrupt Service Routine (PTMIS) responds to interrupts from the timer, extracts the next count computed by CTS routine from the timing buffer and outputs it to the timer. Communication between the two routines requires two timing buffers. The CTS routine fills one buffer with the new timing sequence while the Pulse Timer Module Interrupt routine gets its counts from the other buffer. At the 60° segment boundary the timer controller routine switches to the newly completed buffer and flags the CTS routine to begin a new calculation. Figures 6 and 7 contain flow charts of the two routines.

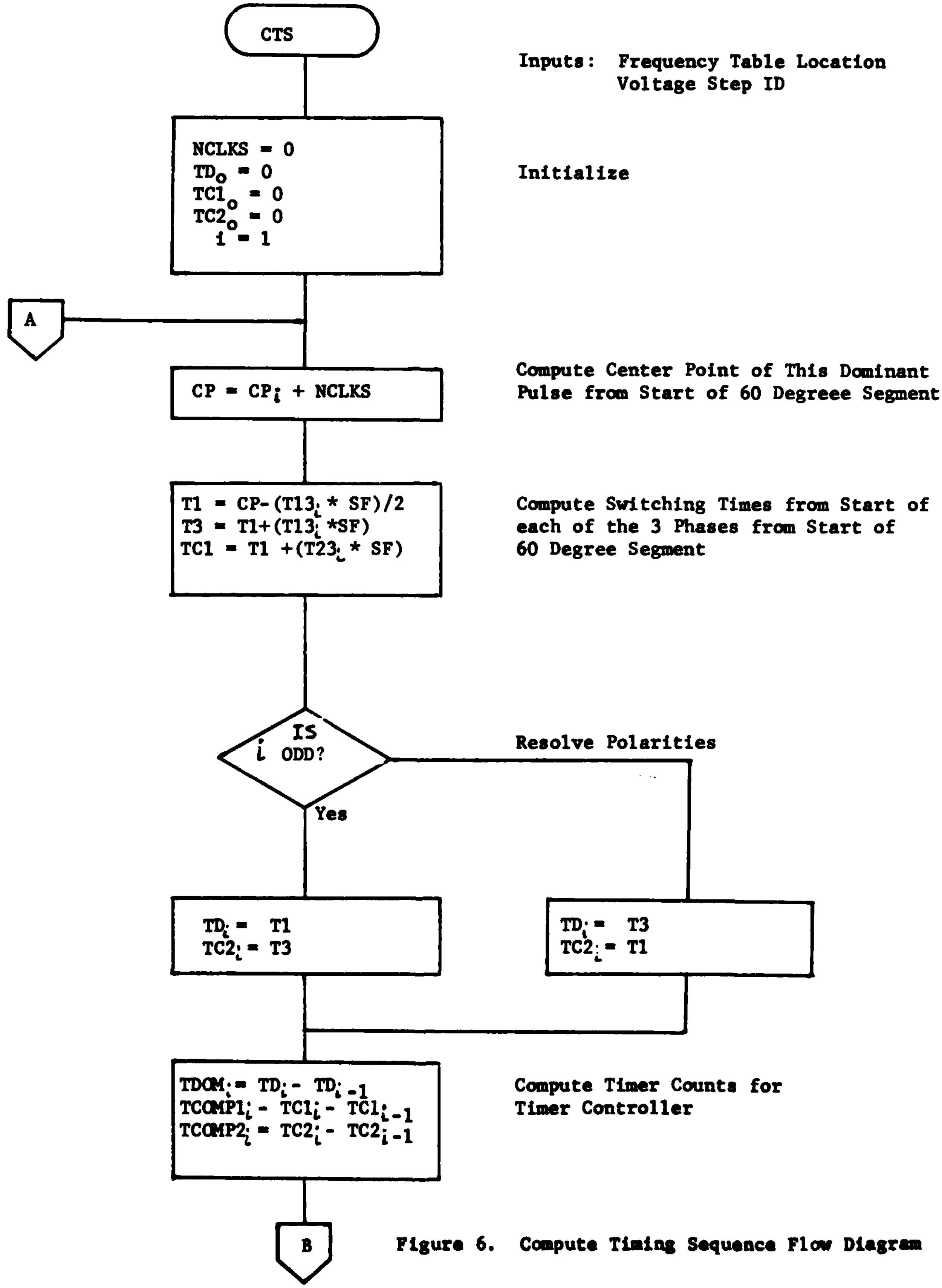


Figure 6. Compute Timing Sequence Flow Diagram

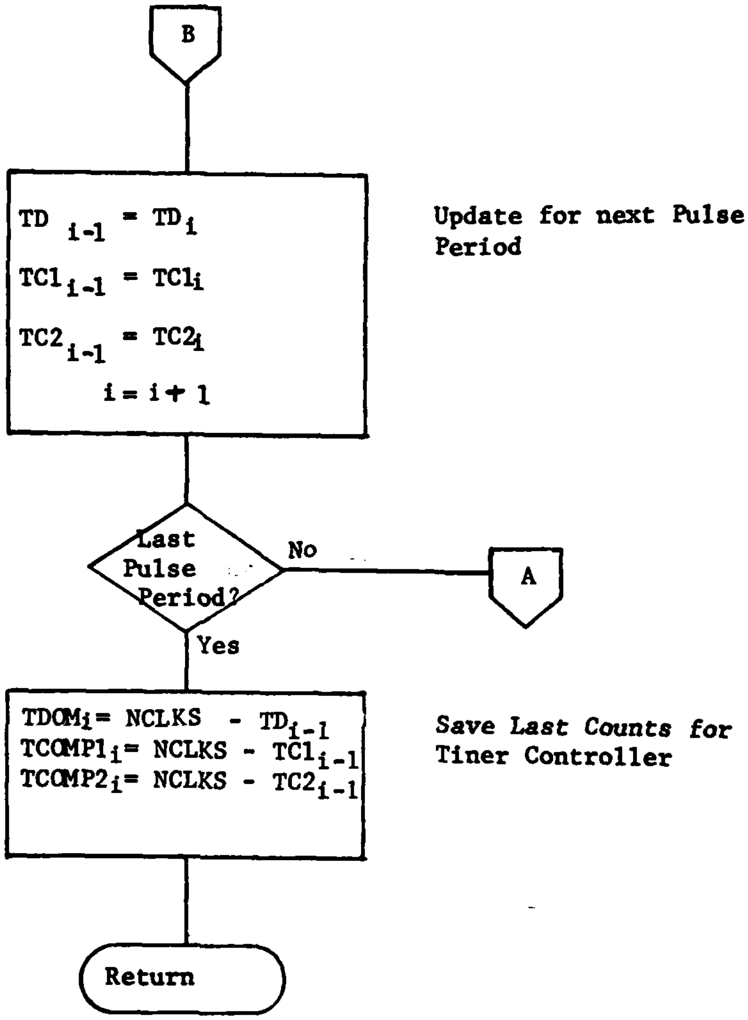


Figure 6. (cont.)

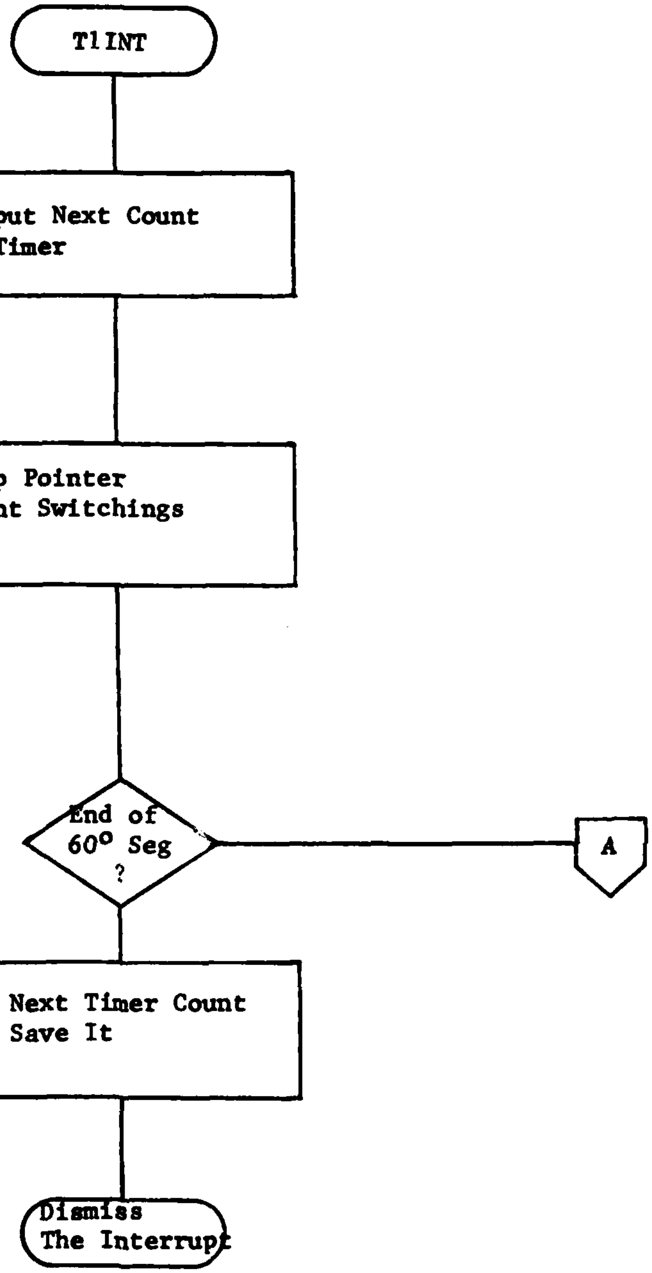


Figure 7. Time #1 Interrupt Service Routine Flow Diagram

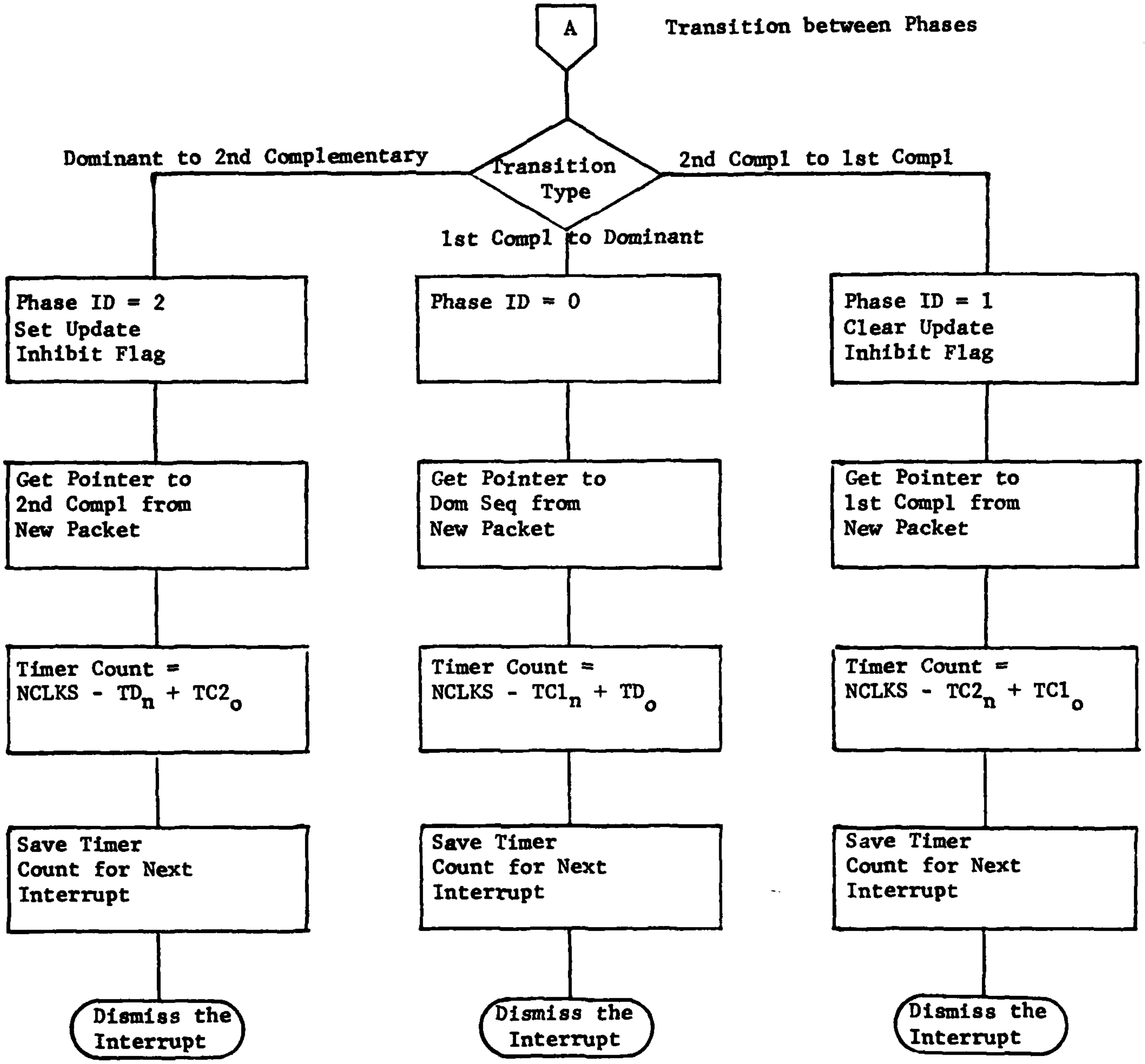


Figure 7. (cont.)

Harmonic Analysis

An harmonic analysis of the PWM waveform was performed in order to arrive at an optimum pulse train within the constraints of the modulation scheme. Harmonic content was traded off against the number of PWM steps and the position of the power-on portion within each step.

Computer programs were created to aid in the analysis and tradeoff study. These programs were written in BASIC and were run on both a Hewlett Packard 9830 and on CDC time sharing service.

The set of pole switching times used in the first portion of the project were derived from the traditional PWM scheme of carrier frequency triangle wave coincidence with the (sine wave) modulation waveform. This scheme has some weaknesses, which are shown below.

Let f(x) be the modulating waveform, and let the triangle wave have a frequency higher than f(x). Then Figure 8 represents one carrier cycle.

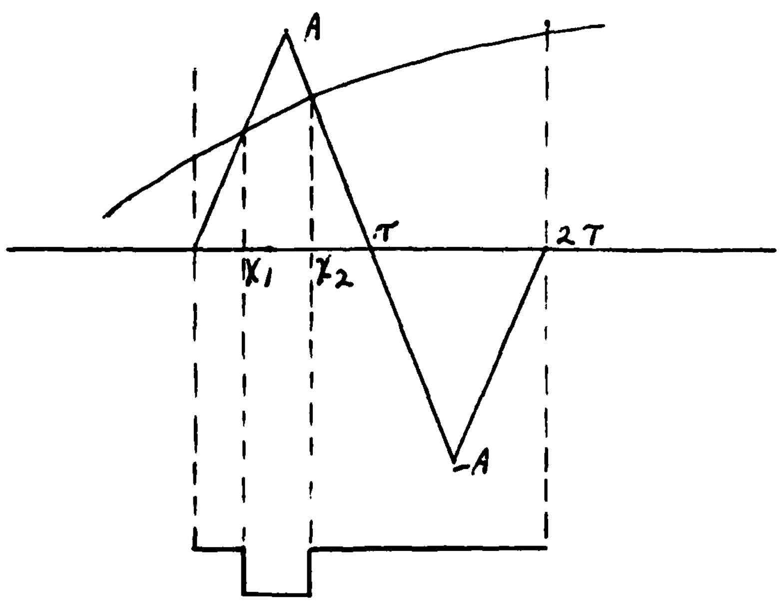


FIG. 8

[1] $$f(X_1) = 2A\left(\frac{X_1}{T}\right)$$

[2] $$f(X_2) = 2A\left(1 - \frac{X_2}{T}\right)$$

OR $$X_1 = f(X_1)\left(\frac{T}{2A}\right)$$

$$X_2 = \left(\frac{T}{2A}\right)\left[2A - f(X_2)\right]$$

[3] DUTY CYCLE AVERAGE, $D = \dfrac{T - (X_2 - X_1)}{2T}$

OR $$D = \frac{f(X_2) + f(X_1)}{4A}$$

which is the average value of f(x) from X_1 to X_2 using a trapezoidal approximation, but taken over the entire period of $2T$. That is, it is the average value of a straight line between $f(x_1)$ and $f(x_2)$, taken as if $f(x_1)$ and $f(x_2)$ were at the segment ends. Because this method extends the slope of its straight line approximation over the entire integration interval, but bases it only on the value of f(x) at X_1 and X_2, it yields values higher than the actual average for curves with negative second derivatives and lower than average for curves with positive second derivatives.

For carrier frequencies much greater than the modulating frequency this PWM scheme is generally acceptable due to its ease of implementation in an analog system. In a digital system, however, such as the one under development here, freedom to choose switching times without impacting the complexity of the modulation scheme allows more accurate output waveform synthesis.

Since the motor acts as an integrating filter, it is desirable to generate the PWM such that the duty cycle of each step is proportional to the integral of the output sine wave over the step period. A computer program was used to generate these integral values for all the possible numbers of steps to be considered. The results of the program, which computes

$$-G(X) = \frac{\mathrm{COS}(X1) - \mathrm{COS}(X2)}{(X1-X2}\,,$$

appears in Appendix D.

The harmonic content of the output wave is of concern in two areas. First, the odd harmonics must be sufficiently suppressed for efficient forward operation of the motor. Second, the output must be easily conditioned to provide clean waveforms for flux calculations.

A computer program was written to generate the desired PWM waveform for any combination of the carrier and output frequencies of interest, and then perform a harmonic analysis on the PWM. The on portion within each step may be chosen as beginning at the start of the step or as being centered within the step. Program output includes harmonic amplitudes relative to the fundamental. For the runs shown, levels below one percent were suppressed from the printout for clarity, although any printing threshold may be chosen when using this program. The program and its output appears in Appendix C.

Plotting and analyzing this output yielded a set of PWM steps per output cycle for each discrete output frequency from seven to sixty hertz. Initially all possibilities of steps per output cycle were permitted in order to learn what optimum combinations existed. Then, due to the mechanization of the modulation scheme in the microprocessor, only odd multiples of three steps per output cycle

were allowed in the analysis. The elimination of some of the even numbers of steps caused large jumps in carrier frequency at the higher output rates. This effect was eased somewhat by shifting all the carriers up to an average of about 700Hz from the previous 540 Hz. See Figures 9 and 10.

One goal in the manipulation of the harmonic data was to choose a combination which keeps unwanted frequencies in the flux loop input down at least 30 dB with only simple filtering. With the step/output combinations finally chosen, this goal was met with a 70 Hz pole low pass filter. The design and response of the test filter appear in Figures 11 and 12.

As a first test of the harmonic analysis and modulation scheme, one phase of the processor output was interfaced with a breadboard filter and a 540 Hz carrier with 36 Hz modulation was transmitted through it. The resultant sine wave observed on an oscilloscope looked good and was delayed about 4 ms as expected. Amplitude control with the processor was exercised satisfactorily. The 540 Hz carrier was used since values for it had been previously stored in the processor memory. Photographs of the oscilloscope observations appear in Figure 13. Carrier frequencies under 500 Hz produced unacceptably distorted output waveforms. Inspection of the waveform with only 9 steps per cycle chosen by the triangular-intercept method showed distortion due to the errors of that scheme.

Later, the one-third horsepower three phase motor was operated by processor generated signals through a motor driver and the phase A current monitored. Figure 14 shows photographs of the sensed current, both filtered for processing and unfiltered. These waveforms confirm the choice of PWM parameters.

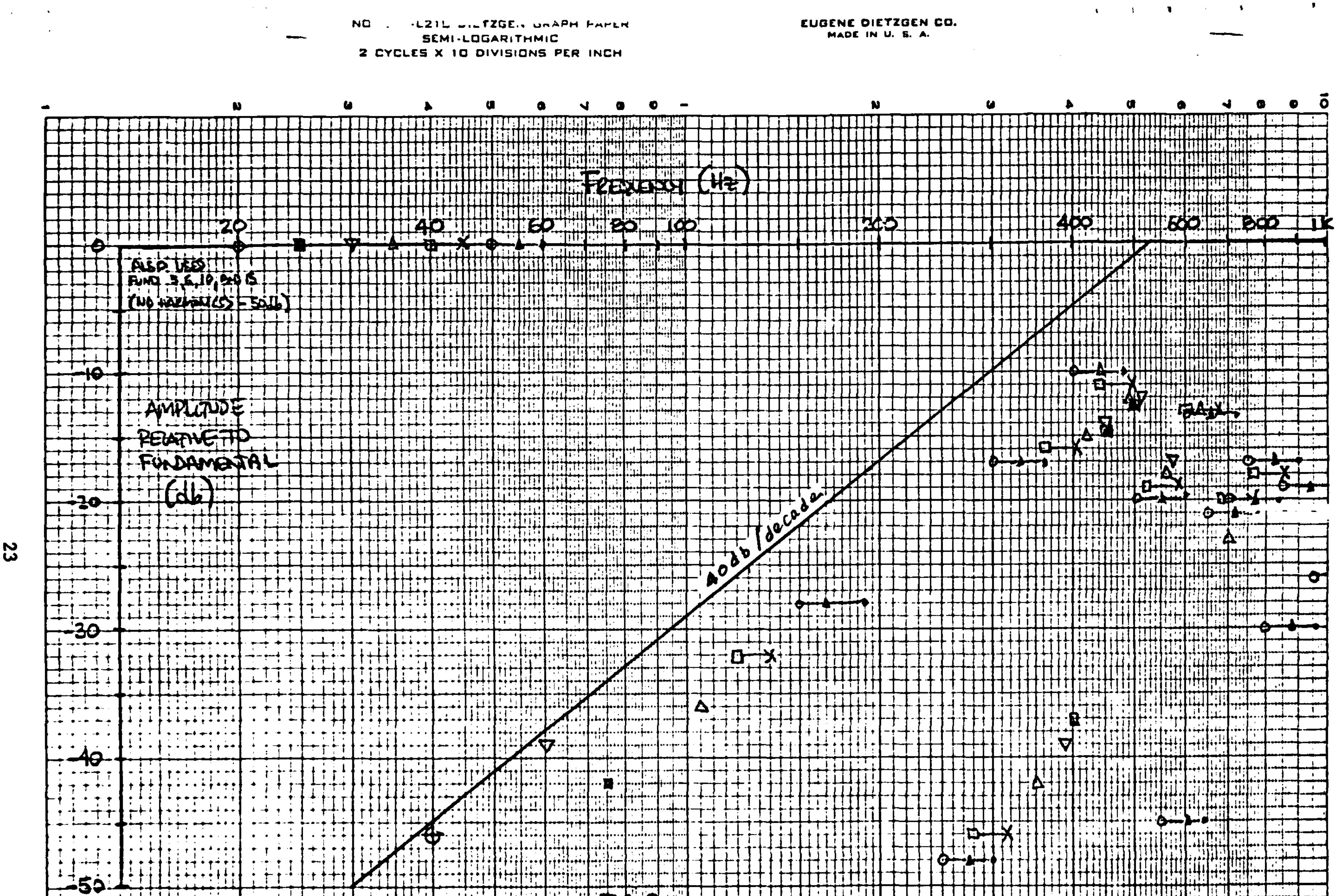


FIG. 9
HARMONIC COMPONENTS ABOVE 50db BELOW THE FUNDAMENTAL FOR OUTPUT FREQUENCIES FROM 5 TO 60 Hz. (SEE ALSO PRINTED DATA)

DBZ
12/21/78

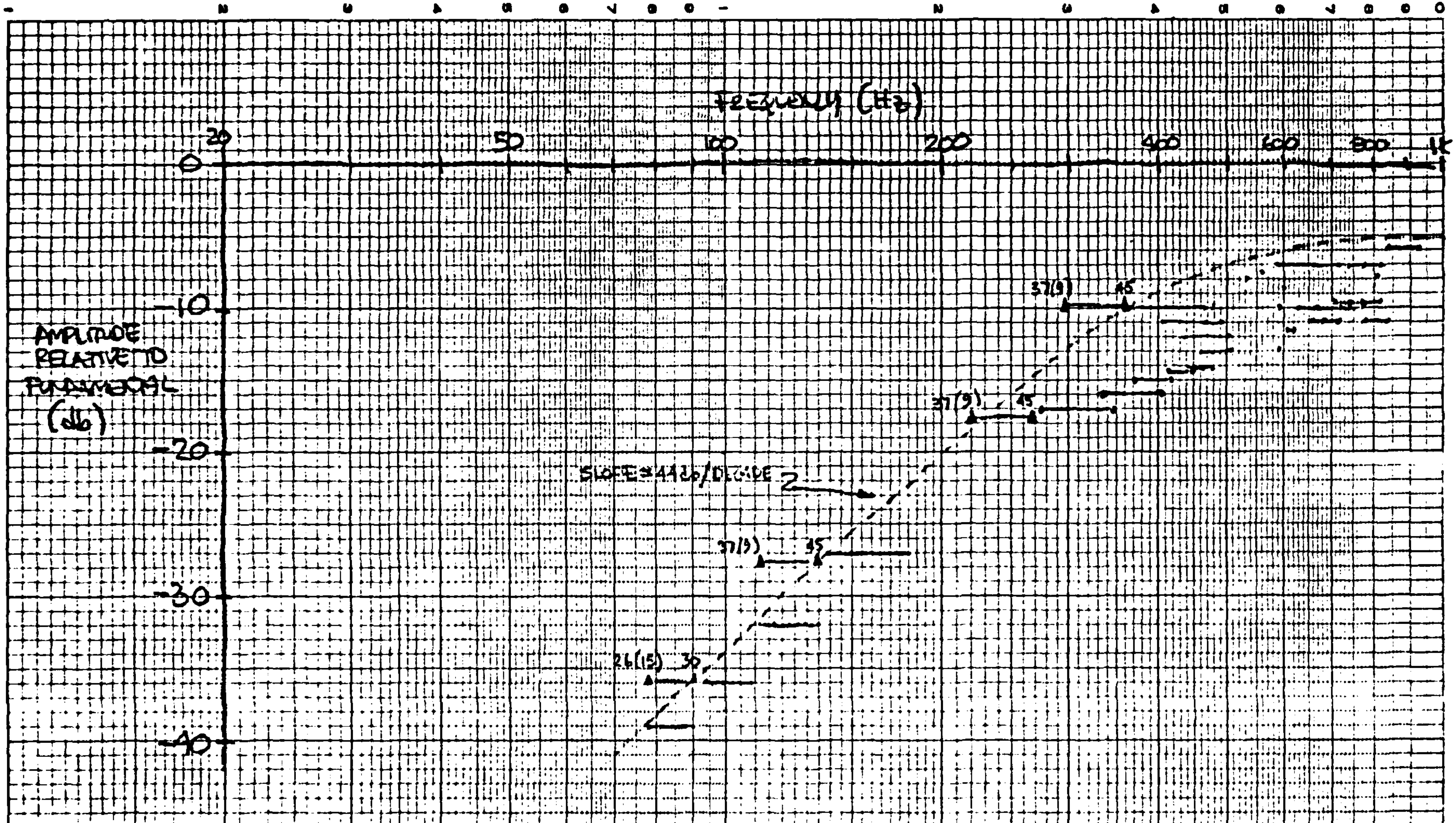
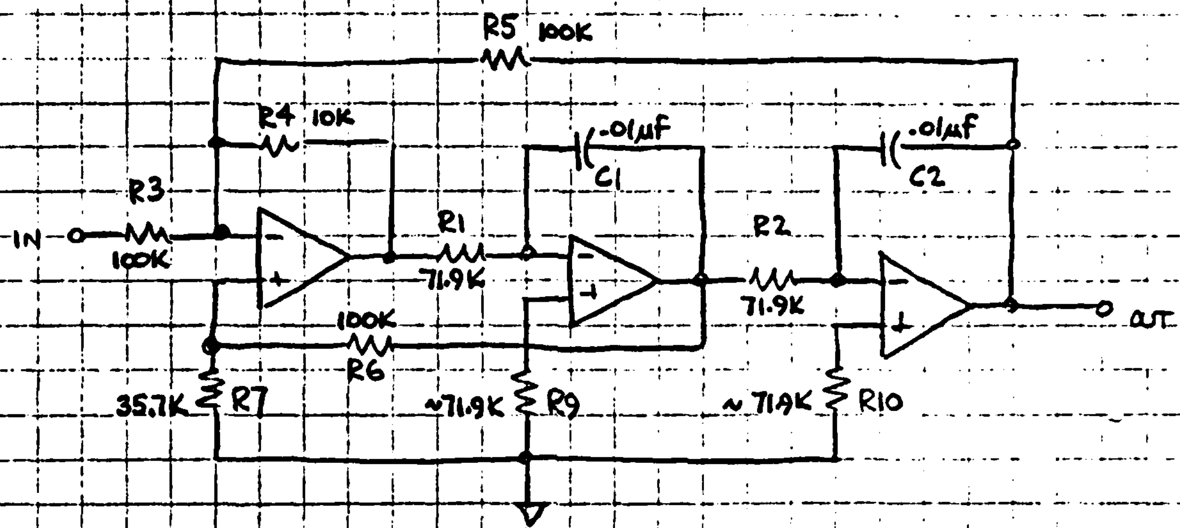


FIG. 10
RESULTS OF HARMONIC ANALYSIS OF
ALL INTERLOCK OUTPUTS 7-60 Hz
(SIGNIFICANT HARMONICS)

▲ USING ODD MULT OF 2 SINE ... 3(4)

FIG. 11 2 POLE LPF, CRITICALLY DAMPED; $f_c = 70\,HZ$ DESIGN

"UNIVERSAL-STATE VARIABLE" CONFIGURATION, INVERTING LPF MODE.



① CHOOSE: $\boxed{\begin{array}{l} R5/R4 = 10 \\ R4 = 10K \\ R5 = 100K \\ R3 = 100K \\ R6 = 100K \end{array}}$

②
$$R1 = R2 = \left(\frac{1}{\omega_0}\right)\sqrt{\frac{R4}{R5 \cdot C1 \cdot C2}}$$

CHOOSE $\boxed{C1 = C2 = .01\mu f}$

THEN
$$R1 = R2 = \left(\frac{1}{2\pi \times 70}\right)\sqrt{\frac{.1}{(.01 \times 10^{-6})^2}} = \boxed{71.9K\Omega = R1 = R2}$$

③
$$R7 = \frac{100K\Omega}{3.8Q - 1} \quad \text{(FOR UNITY GAIN)}$$

FOR CRITICAL DAMPING $Q = 1$

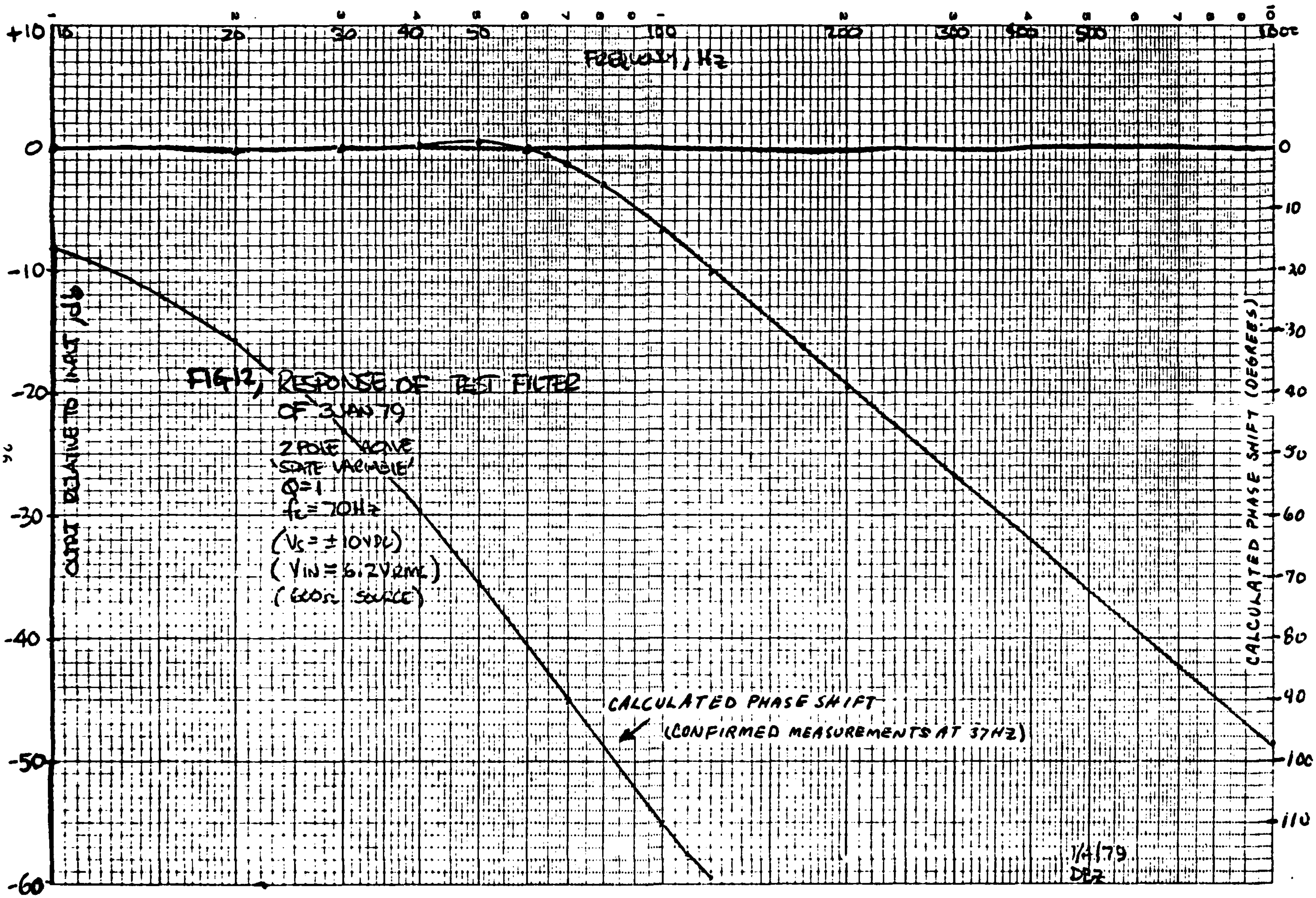
SO
$$R7 = \frac{100K}{2.8} = \boxed{35.7K\Omega = R7}$$

④ $\boxed{R_{10} = R9 = R1 = 71.9K, \text{ USE } 68K}$

DEF 1/2/79

NO. 341-L210 DIETZGEN GRAPH PAPER
SEMI-LOGARITHMIC
2 CYCLES X 10 DIVISIONS PER INCH

EUGENE DIETZGEN CO.
MADE IN U. S. A.

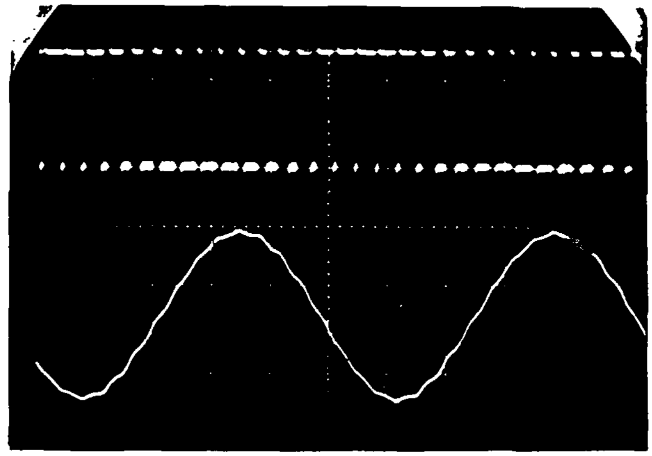


• PWM Waveform, 2V/CM

36 Hz Sine Wave Modulation
540 Hz Carrier Horiz.
Scale: 5 ms/CM

• Filtered Output, 1V/CM

2 Pole 70 Hz Filter



• PWM Waveform, 2V/CM

36 Hz Sinewave Modulation
540 Hz Carrier Horiz.
Scale: 2 ms/CM

• Filtered Output, 1V/CM

2 Pole 70 Hz Filter

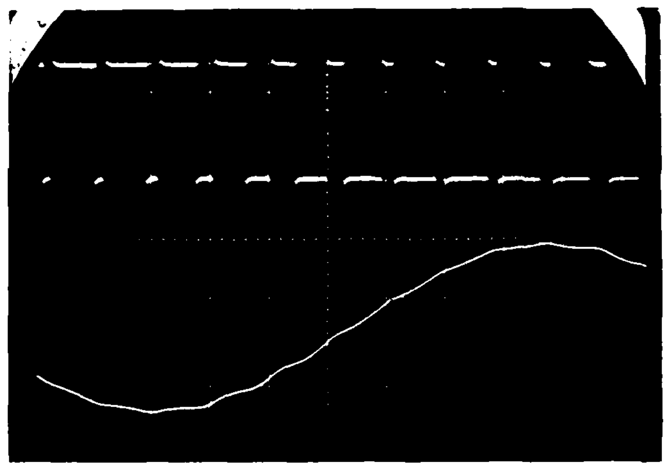


Figure 13 PWM WAVEFORMS

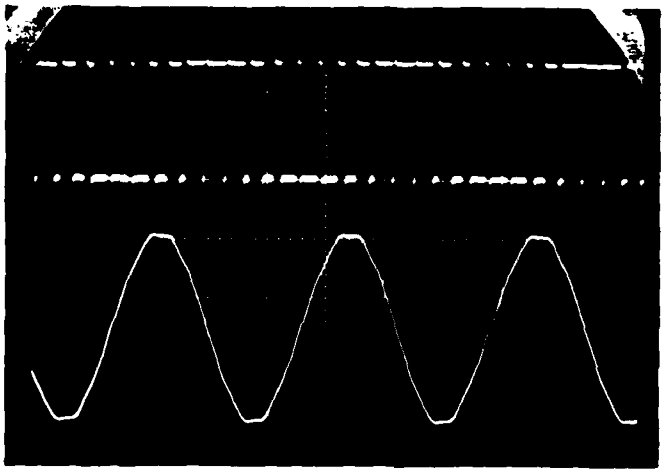
• PWM Waveform, 2V/CM

60 Hz Sine Modulation
540 Hz Carrier Horiz.
Scale: 5 ms/CM

• Filtered Output, 1V/CM

2 Pole 70 Hz Filter

(Flat-Topping due to Incorrect
Modulation Pulse Widths)

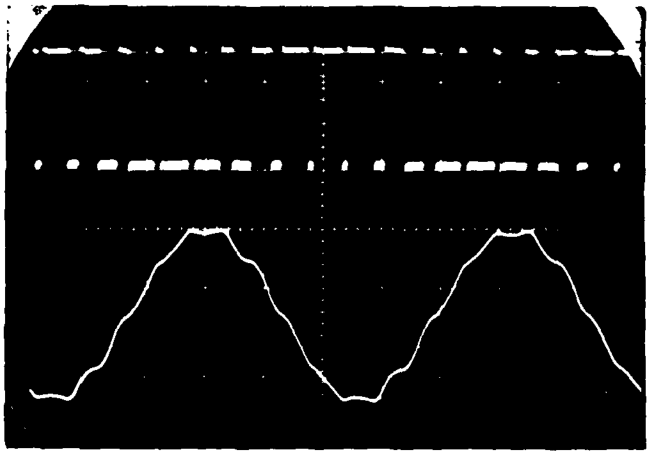


• PWM Waveform, 2V/CM

37 Hz Sine Wave Modulation
333 Hz Carrier Horiz.
Scale: 5 ms/CM

• Filtered Output, 1V/CM

2 Pole 70 Hz Filter



Inadequate number of PWM pulses
per modulation cycle (low carrier
frequency) causes roughtoutput waveform.

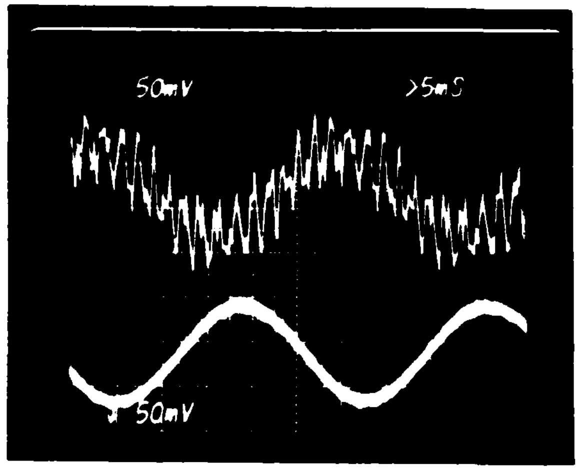
Figure 13 PWM WAVEFORMS
(Continued)

- ØA Motor Current, 0.5A/CM

  36 Hz Sine Wave Modulation
  540 Hz Carrier Horiz.
  Scale: 5 ms/CM

- Filtered Current Sense, 0.5A/CM

  2 Pole 70 Hz Filter

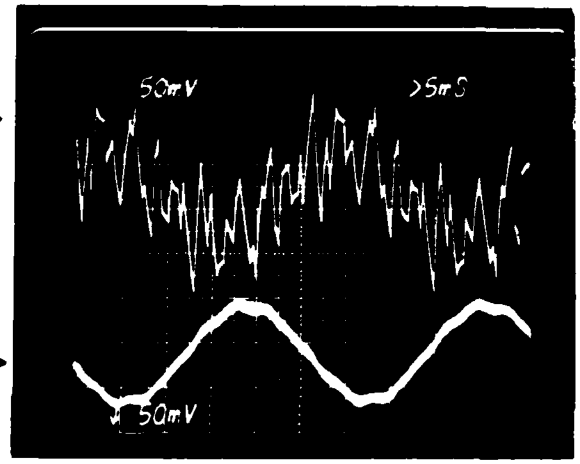


- ØA Motor Current, 0.5A/CM

  37 Hz Sine Wave Modulation
  333 Hz Carrier Horiz.
  Scale: 5 ms/CM

- Filtered Current Sense, 0.5A/CM

  2 Pole 70 Hz Filter



- ØA Motor Current, 0.5A/CM

  60 Hz Sine Wave Modulation
  540 Hz Carrier Horiz.
  Scale: 5 ms/CM

- Filtered Current Sense, 0.5A/CM

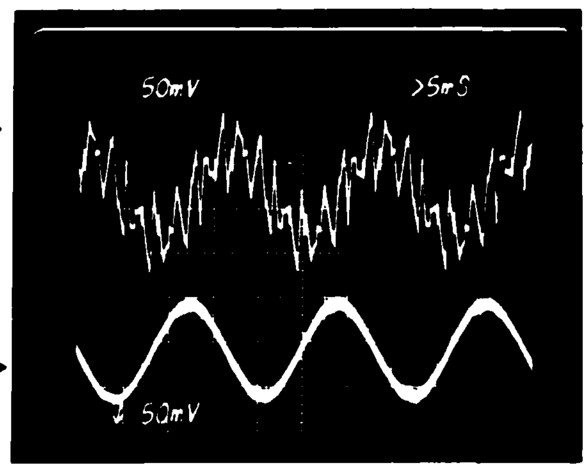


Figure 14 MOTOR WAVEFORMS

SYSTEM OPERATION

Figure 15 is a block diagram of the overall system. In the complete system, sensor conditioners operate on the current transformer and voltage probe inputs to provide analog voltages proportional to each of the three phase currents and voltages. I_B and I_C are added differentially and scaled, as are V_A and V_B. The four functions existing after the scaling and addition processes are then filtered. Each filter is identical so as to impose the same delay in each signal path. Thus, the phase relation among the d-q variables resulting at the output of the filters is preserved. Each variable is converted to a digital signal with an 8 BIT A/D converter and is transmitted to the microprocessor through a PIA.

The microprocessor and a motor driver were interfaced and successfully operated a three phase one-third horsepower motor over three pre-programmed speeds and many amplitudes.

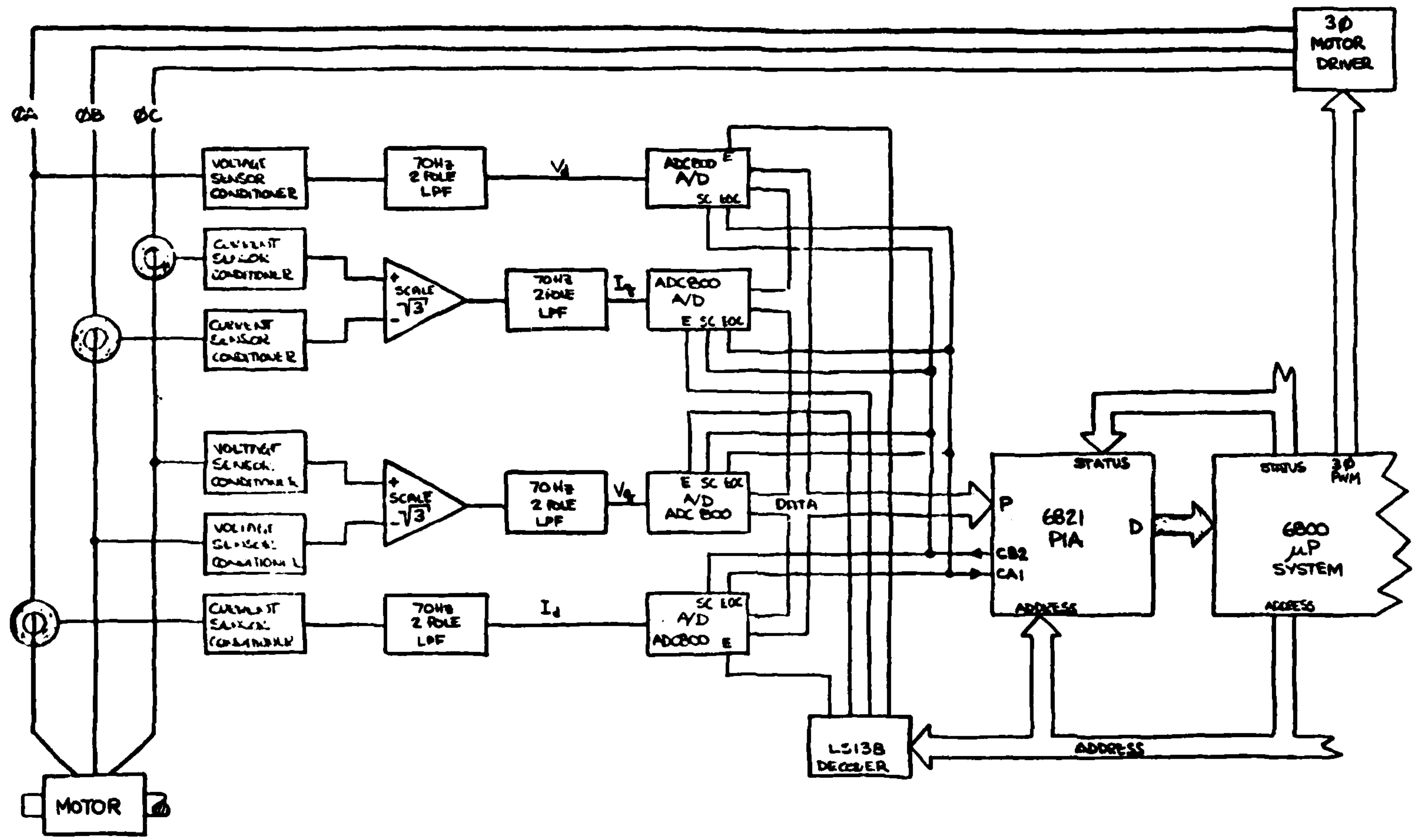


FIGURE 15
μP CONTROLLED AC MOTOR DRIVE
OVERALL BLOCK DIAGRAM

Ⓤ DBZ 1-18-79

Appendix A

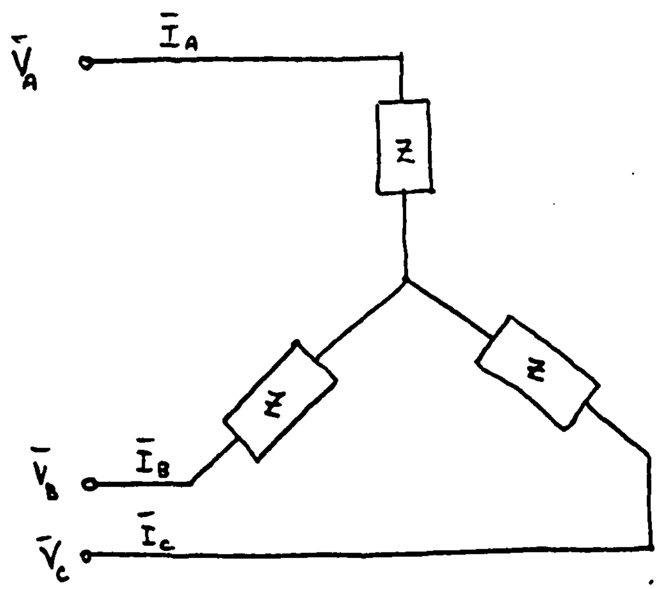
DERIVATION OF AIR GAP FLUX SYNTHESIS SIGNAL

Let applied voltage $\bar{V} = V \sin \omega t$ and then consequent current $\bar{I} = I \sin(\omega t - \phi$

and the reactive power in one phase

$$W_x = \tfrac{1}{2} V I \sin \phi$$

For a 3-phase system



$$\bar{V}_A = V \sin \omega t$$

$$\bar{V}_B = V \sin(\omega t - 2\pi/3)$$

$$\bar{V}_C = V \sin(\omega t + 2\pi/3)$$

$$\bar{I}_A = I \sin(\omega t - \phi)$$

$$\bar{I}_B = I \sin(\omega t - \tfrac{2\pi}{3} - \phi)$$

$$\bar{I}_C = I \sin(\omega t + \tfrac{2\pi}{3} - \phi)$$

$$Z = \frac{V}{I}\left(\cos\phi + j \sin\right.$$

FiG. 1a

The three phase system can be expressed in terms of d - q (direct and quadrature) variables.

Three phase vector diagram:

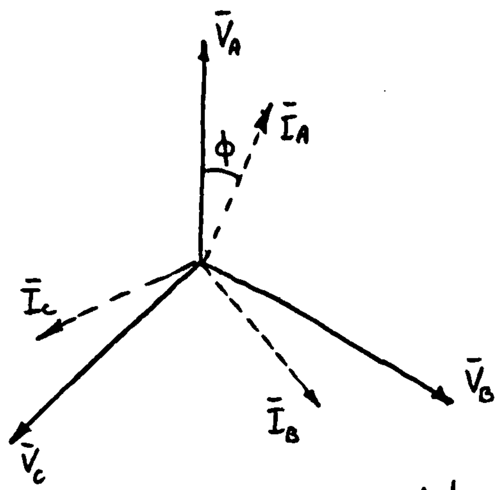
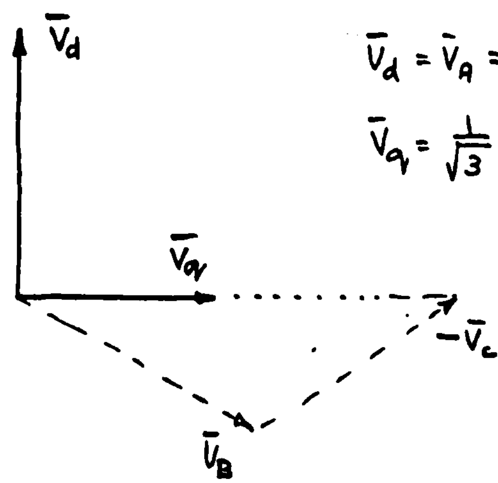


FiG. 1 b

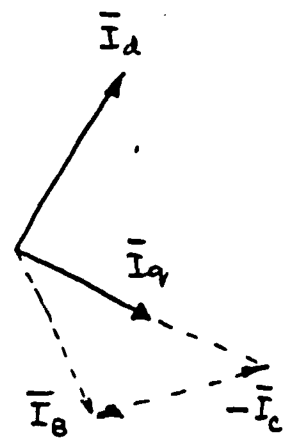
d - q equivalent vector diagram:



$$\bar{V}_d = \bar{V}_A = V \sin \omega t$$

$$\bar{V}_q = \frac{1}{\sqrt{3}}\left(\bar{V}_B - \bar{V}_C\right) = -V \cos \omega t$$

FiG. 1c



$$\bar{I}_d = \bar{I}_A = \bar{I}\sin(\omega t - \phi)$$

$$\bar{I}_q = \frac{1}{\sqrt{3}}\left(\bar{I}_B - \bar{I}_C\right) = -\bar{I}\cos(\omega t - \phi)$$

FIG. 1d

Products can be formed for $\bar{V}_d \bar{I}_q$ and $\bar{V}_q \bar{I}_d$ as follows:

$$\bar{V}_d \bar{I}_q = -VI \sin \omega t . \cos(\omega t - \phi)$$

$$= -\frac{VI}{2} \sin \phi - \frac{VI}{2} \sin(2\omega t - \phi)$$

$$\bar{V}_q \bar{I}_d = -VI \cos \omega t . \sin(\omega t - \phi)$$

$$= \frac{VI}{2} \sin \phi - \frac{VI}{2} \sin(2\omega t - \phi)$$

Each is composed of a DC term and an AC term at twice the frequency of the applied voltage. Taking the difference of the two terms doubles the DC component and cancels the AC components

$$\bar{V}_q \bar{I}_d - \bar{V}_d \bar{I}_q = VI \sin \phi$$

which can be seen to be twice the reactive power, Wx, in one phase. Thus, the first objective of the flux synthesis technique is to generate the d-q variables from the three phase currents and voltgaes. This is shown in the block diagram as follows:

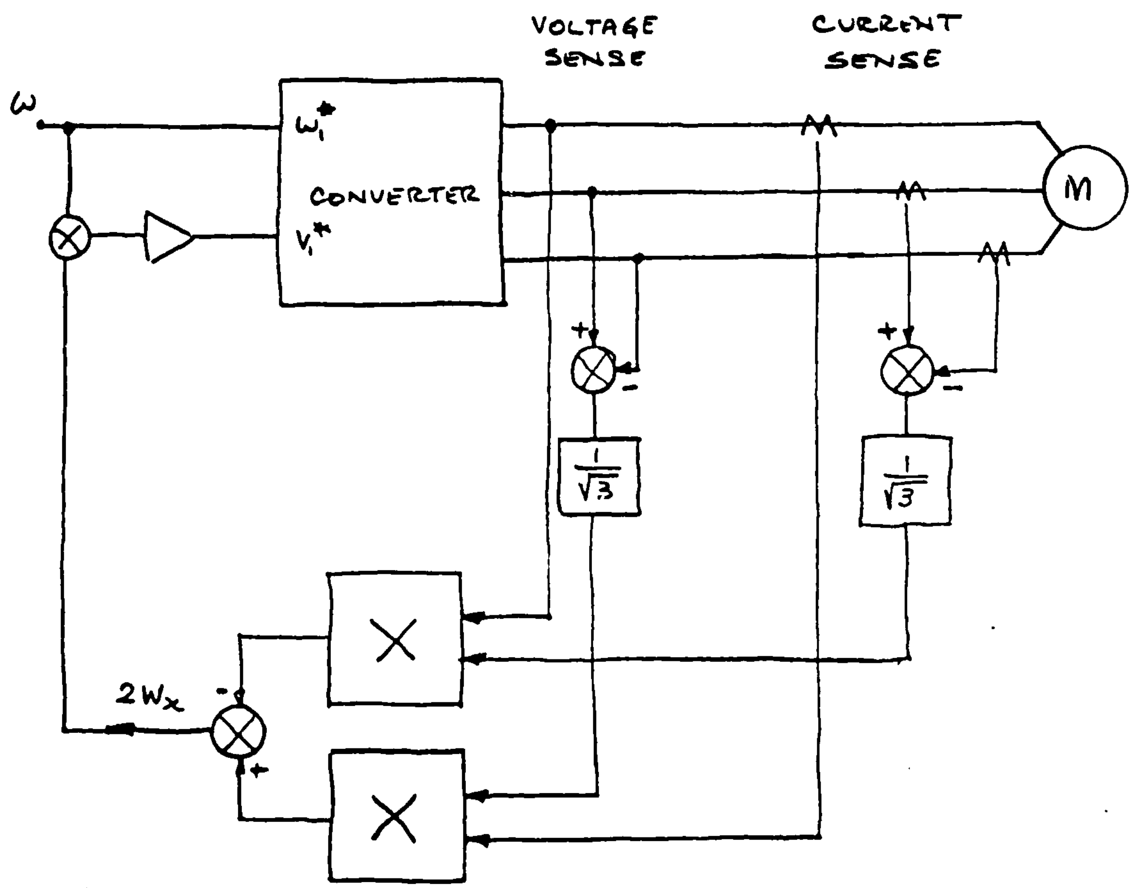


FIG. 2a

Thus it is possible to generate a signal proportional to the reactive power, Wx, by measuring only the voltages and currents at the motor terminals.

Since the real objective is to maintain the component of W_x due to L_m flowing in L_m, W_x must be broken down into its component terms.

Referring to the equivalent circuit (Figure 2b)

$$W_x = \frac{1}{2}\bar{I}_1^2 \omega_1 L_1 + \frac{1}{2}\bar{I}_m^2 \omega_1 L_m + \frac{1}{2}\bar{I}_2^2 \omega_1 L_2$$

The first term is a function of the measured current, a fixed frequency and fixed inductance and is known. The second term is the desired component of W_x and the third term is a term that is undefined unless $\bar{I}_2$ can be accounted for.

Referring to the vector diagram of Figure 2c

$$\bar{I}_1^2 = (I_M + I_{2x})^2 + I_{2Y}^2$$

$$= I_M^2 + 2 I_m I_{2x} + I_2^2$$

but $$\bar{I}_{m1} = \frac{E_1}{\omega_1 L_m}$$

$$\bar{I}_{3x} = \bar{I}_2^2 \frac{\omega_1 L_2}{E_1}$$

$$\therefore \quad \bar{I}_1^2 = I_M^2 + 2\bar{I}_2^2 \frac{L_2}{L_m} + \bar{I}_2^2$$

$$\bar{I}_1^2 - \bar{I}_m^2 = \frac{L_m + 2L_2}{L_m} \cdot \bar{I}_2^2$$

which allows substitution for $\bar{I}_2^2$ in terms of $\bar{I}_1^2 - \bar{I}_M^2$.

$$W_x = \frac{1}{2}\omega_1 L_1 \bar{I}_1^2 + \frac{1}{2}\bar{I}_m^2 \omega_1 L_m + \frac{1}{2}\frac{L_m}{L_m + 2L_2}\omega_1 L_2 (\bar{I}_1^2 - \bar{I}_m^2)$$

$$2W_x = \omega_1 L_1 \bar{I}_1^2 + \omega_1 L_m \frac{L_m + L_2}{L_m + 2L_2} \cdot \bar{I}_m^2 + \omega_1 \frac{L_m L_2}{L_m + 2L_2} I_1^2$$

$$W_x = \frac{\omega_1}{2(L_m + 2L_2)}\left[(L_m^2 + L_m L_2) I_m^2 + (L_m L_2 + L_m L_1 + 2L_1 L_2) I_1^2\right]$$

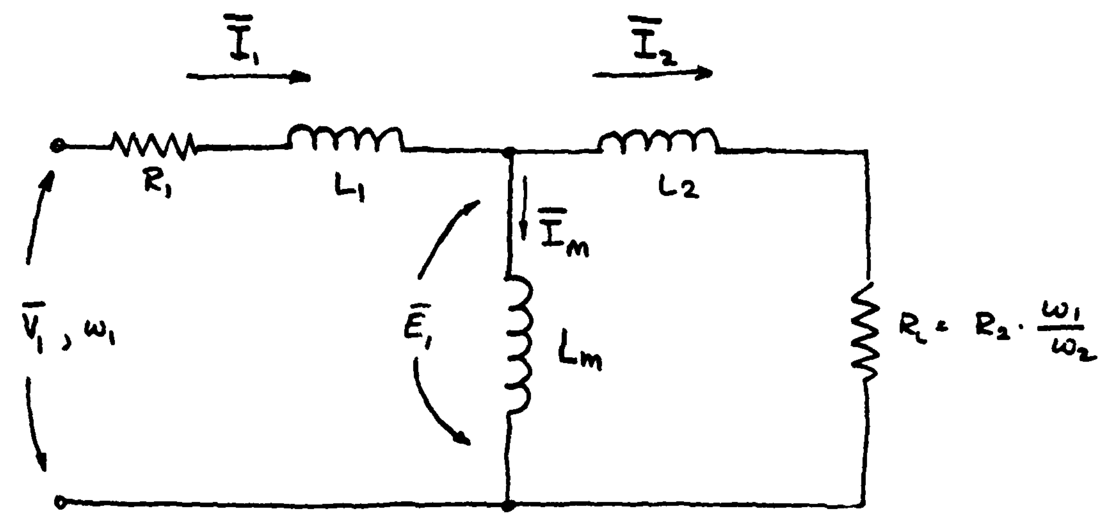
i.e., W_x is a function of the derived I_m and known terms L_m, L_2, L_1 and W_1. Thus, by operating on W_x appropriately I_m and hence the air gap flux can be determined.

Rearranging the previous equations we have:

$$\frac{L_m (L_m + L_2)}{2(L_m + 2L_2)}\omega_1 I_m^2 = W_x - \frac{1}{2}\omega_1 I_1^2 \left(L_1 + \frac{L_m L_2}{L_m + L_2}\right)$$

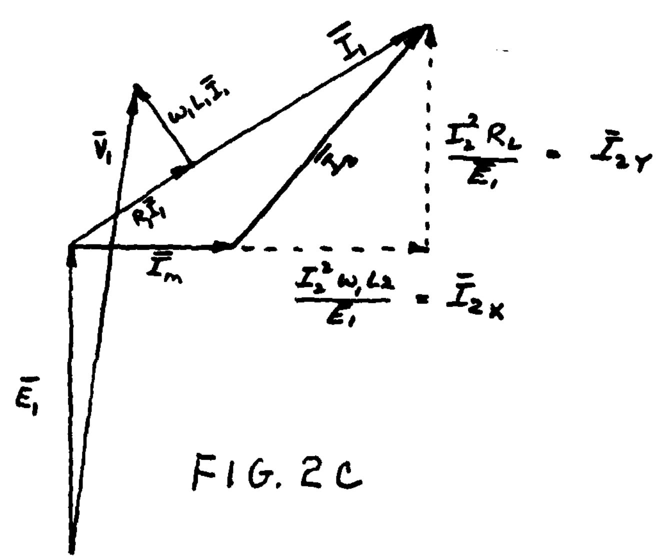
$$= W_x - \frac{1}{2} I_1^2 \cdot \omega_1 \left(L_1 + \frac{L_m L_2}{L_m + L_2}\right)$$

i.e. $$K_1 \omega_1 I_m^2 = W_x - \frac{1}{2} I_1^2 \omega_1 L'$$



$\bar{I}_1$
$\bar{I}_2$
R_1
L_1
L_2
$\bar{I}_m$
$\bar{V}_1, \omega_1$
$\bar{E}_1$
L_m
$R_L = R_2 \cdot \frac{\omega_1}{\omega_2}$

FIG. 2 b



$\bar{I}_1$
$\omega_1 L_1 \bar{I}_1$
$\bar{V}_1$
$\frac{I_2^2 R_L}{E_1} = \bar{I}_{2Y}$
$\bar{I}_m$
$\frac{I_2^2 \omega_1 L_2}{E_1} = \bar{I}_{2X}$
$\bar{E}_1$

FIG. 2C

from which a revised block diagram can be drawn to allow for the required reduction in Wx by a term proportional to the reactive power that would be developed in an inductor $L_1 + LmL_2/(Lm + L_2)$ if the stator current were to flow through it. To compute $W_x - \frac{1}{2} I_1^2 \omega_1 L'$ we can reduce the voltage components of Wx by an amount equivalent to the voltage drop across L' due to $\bar{I}_1$ flowing. A simplified equivalent circuit is as follows:

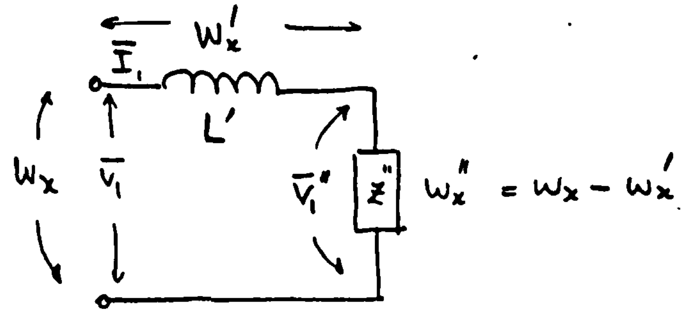


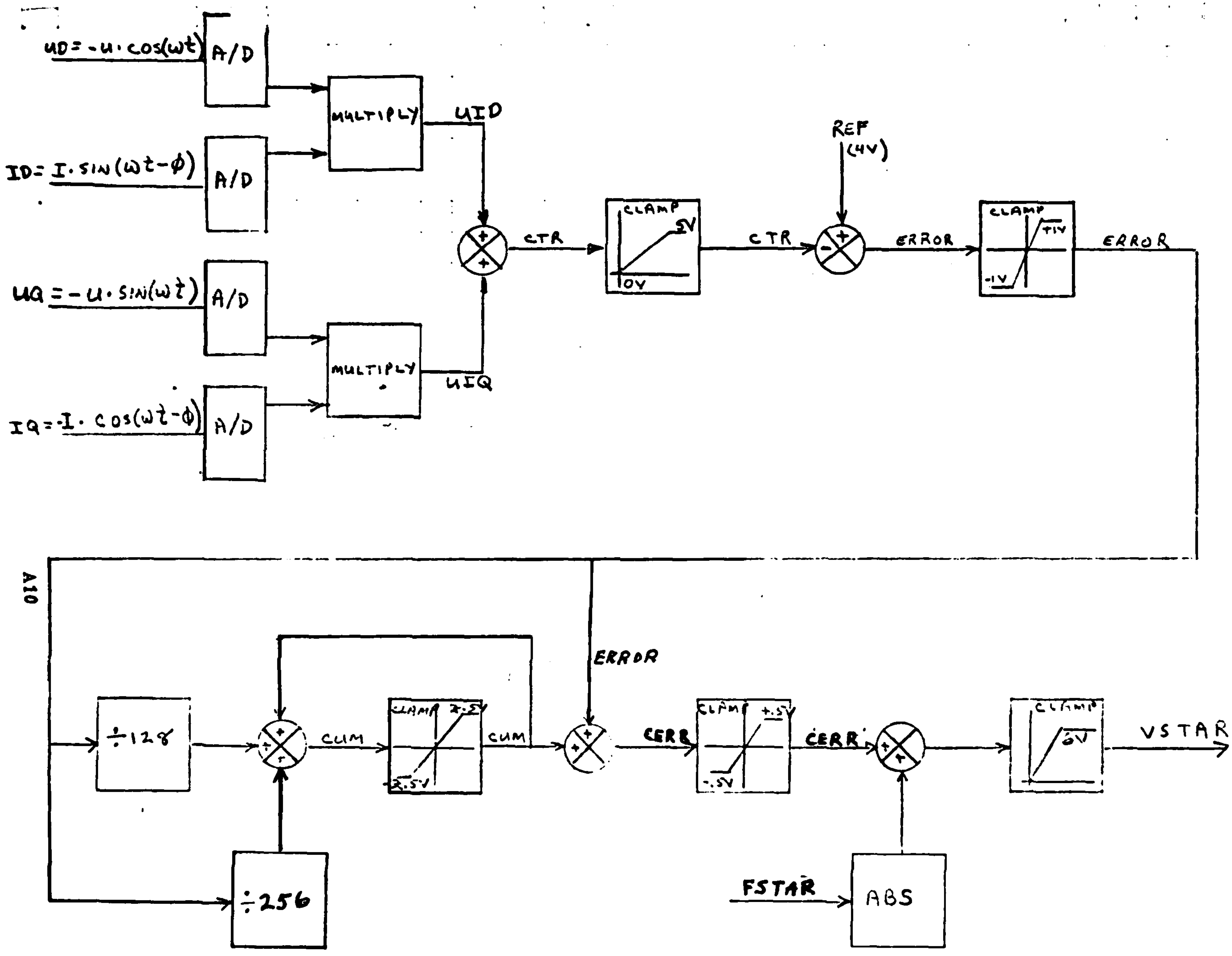
FIG.2d

The feedback function is to keep $\bar{I}_m$ constant for any value set for W. It does this by modifying the convertor output voltage which in turn causes a change in $\bar{I}_m$. Since this is a closed loop feedback involving a load with transfer characteristic $\frac{1}{s+T_1}$ where T_1 reflects the $\frac{L}{R}$ time constant of the induction motor the feedback function will need compensation of the form $\frac{s+T_2}{s}$ i.e. an integrating term to bring down the overall gain as frequency increases and a lead term to compensate for the lag introduced by the inductive time constant of the load. This steers the phase angle away from -180 until the gain has become less than unity.

Figure 3 shows a block diagram of a flux regulator loop following the model of Figure 2 i.e. without allowing for the drop in L_1. The scaled inputs V_q, I_d and V_d , I_q are multiplied together and then subtracted to produce $\bar{V}\bar{I}\sin\phi$ (W_x) which is the control signal CTR.

This is kept within bounds by a clamping circuit and is compared with a reference signal. The consequent error signal, again held within limits is fed to a loop which provides the $\frac{S + 2}{S}$ function. The output is again held within limits and is used to modify the voltage control level V * which would normally be kept constant for any given input frequency W*. A microprocessor program for this loop has been assembled and run and is attached.

To account for the voltage drop in L_1 and use the model of Figure 4, a term ($j\omega \frac{k_v}{k_I} L'$) must be subtracted from each voltage input. This adds an extra subroutine in the program to multiply $k_I \bar{I}_q$ and $k_I \bar{I}_d$ by the factor L' and ω_1 and subtract them from Vd and Vq.



A10

FIG. 3 BLOCK DIAGRAM – FLUX REGULATOR LOOP

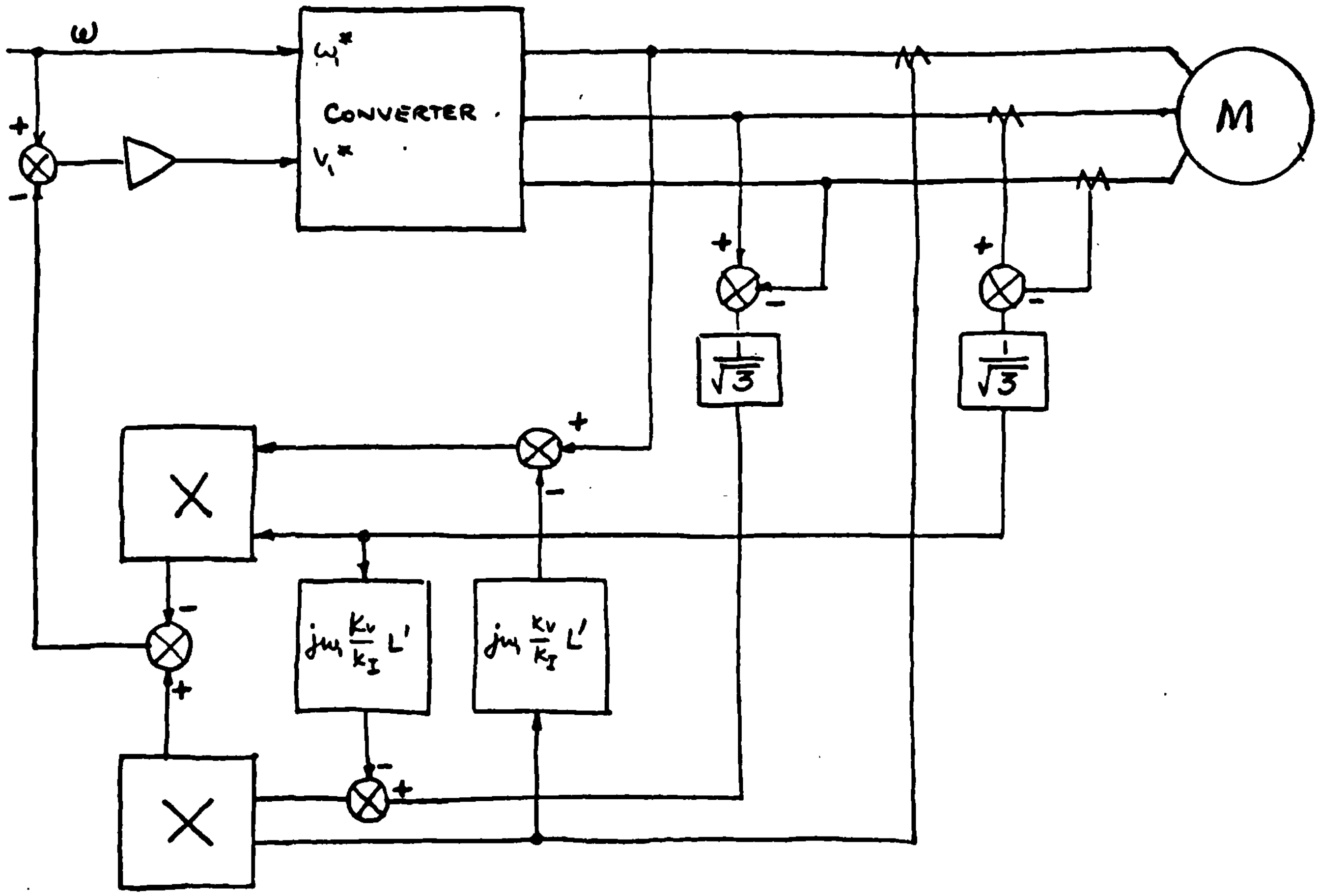

ω
ω_1^*
CONVERTER
V_1^*
M
+
-
$\frac{1}{\sqrt{3}}$
$\frac{1}{\sqrt{3}}$
$j\omega_1 \frac{K_V}{K_I} L'$
$j\omega_1 \frac{K_V}{K_I} L'$

FIG. 4

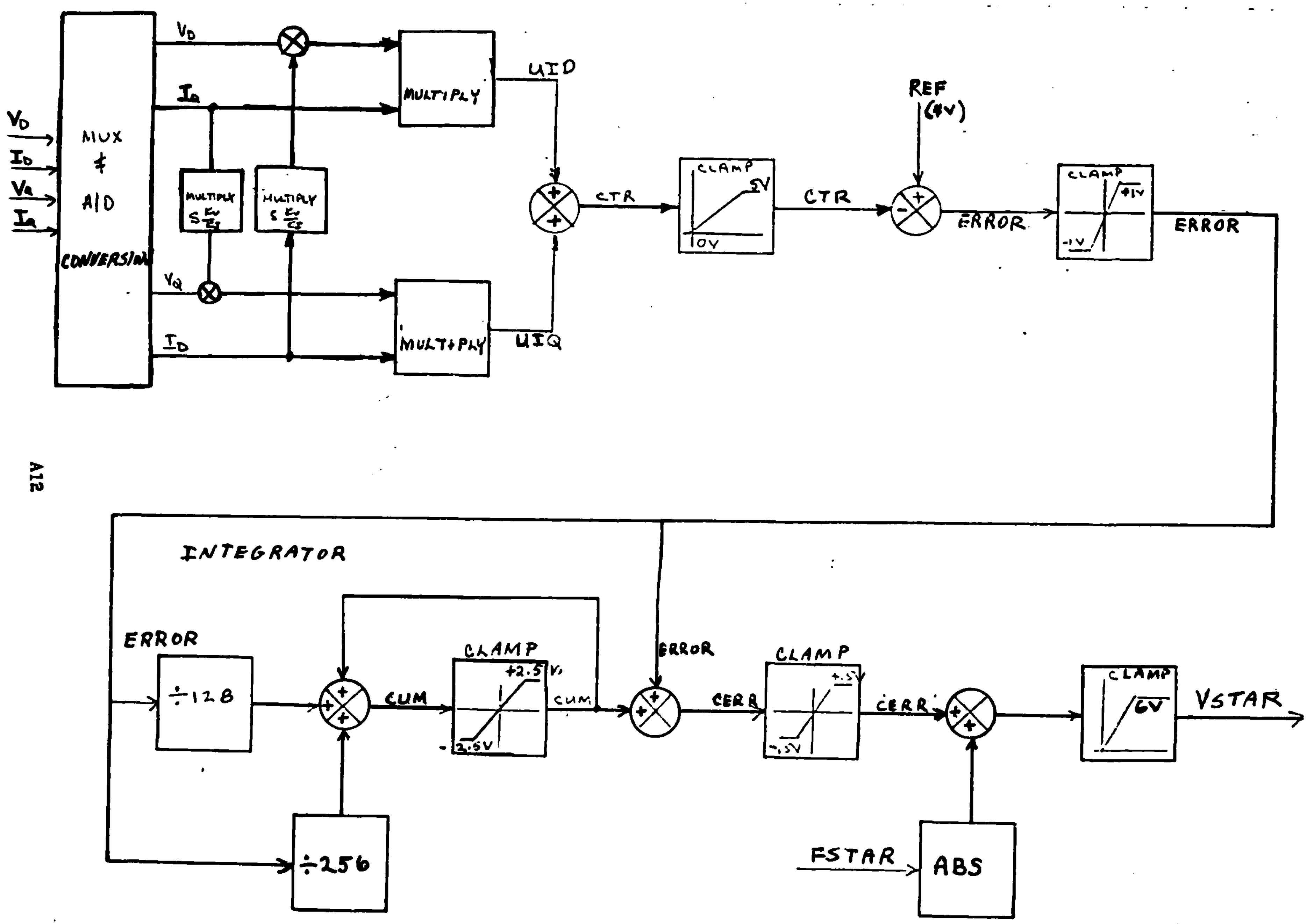


FIG. 5 BLOCK DIAGRAM - FLUX REGULATOR LOOP

Implementing the Integrator

The accumulated error signal (CUM) is defined as:

$$\text{CUM} = \int_0^t 2 \text{ X ERROR } dt$$

This can be computed as:

$$\text{CUM} = \text{CUM} + 2 \text{ X ERROR} \times \Delta T$$

Assuming a flux loop cycle time of .005 seconds the factor 2 X ΔT = .01. In order to take advantage of the binary nature of a computer, where multiplication and division by powers of 2 are merely shift operations, it is convenient to estimate: $2 \text{ X } \Delta T = .01 \simeq \left(\frac{1}{128} + \frac{1}{256}\right)$

The integration equations then become:

$\text{CUM} = \text{CUM} + \frac{\text{ERROR}}{128} + \frac{\text{ERROR}}{256}$, and the ERROR signal is shifted and added as required to compute the error increment for the time interval.

```
                              NAM    FLUX
                         *
                         * FLUX LOOP PROGRAM
                         *
                              OPT    0
     0100                     ORG    $100
                         *
                         *
                         * BLOCK 2.  MULTIPLICATION
                         *           UD*ID; UQ*IQ
                         *
     0100 B6 0400 START       LDA A  UD
     0103 F6 0401             LDA B  ID
     0106 BD 0172             JSR    SMPY      UID TO A,B
     0109 B7 0404             STA A  UID       TEMP SAVE UID
     010C F7 0405             STA B  UID+1
                         *
     010F B6 0402             LDA A  UQ
     0112 F6 0403             LDA B  IQ
     0115 BD 0172             JSR    SMPY      UIQ TO A,B
                         *
                         * BLOCK 3.   CTR=UID+UIQ
                         *
     0118 FB 0405             ADD B  UID+1     ADD LSB'S
     011B B9 0404             ADC A  UID       ADD MSB'S + CARRY
                         *
                         *
                         * BLOCK 4.   ERROR=REF-CTR
                         *            ERROR=CLAMP(ERROR)
                         * (CTR IS IN ACC A)
                         *
     011E 40                  NEG A  -CTR
     011F 8B 66               ADD A  #$66      REF-CTR  (REF=66HEX=4VOLTS)
     0121 C6 1A               LDA B  #$1A      CLAMP FOR ERROR (=1 VOLT)
     0123 BD 018F             JSR    CLAMP     CLAMP ERROR
     0126 B7 0406             STA A  ERROR     SAVE CLAMPED ERROR SIGNAL
                         *
                         *
                         * BLOCK 5.   CUM=CUM+ERROR/128+ERROR/256
                         *            CUM=CLAMP(CUM)   (CLAMP=2.5 V)
                         * (ERROR IS IN ACC A)
                         *
     0129 16                  TAB              ERROR/256
     012A 47                  ASR A  EXTEND    SIGN IN ACC A
     012B 47                  ASR A
     012C 47                  ASR A
     012D 47                  ASR A
     012E 47                  ASR A
     012F 47                  ASR A
     0130 47                  ASR A

     0131 B7 0407             STA A  ER256     TEMP SAVE ERROR/256 (16 BIT
     0134 F7 0408             STA B  ER256+1
                         *
     0137 58                  ASL B  ERROR/128
     0138 49                  ROL A
                         *
     0139 FB 0408             ADD B  ER256+1   ERROR/128+ERROR/256
     013C B9 0407             ADC A  ER256     (16 BIT ADD)
```

```
0137F FB 040A        ADD B  CUM+1      + CUM
0142 B9 0409         ADC A  CUM
                *
0145 F7 040A         STA B  CUM+1      SAVE CUM (LSB'S)
0148 C6 40           LDA B  #$40       CLAMP FOR CUM (= 2.5 V)
014A BD 018F         JSR    CLAMP      CLAMP CUM
014D B7 0409         STA A  CUM        SAVE CUM (16 BITS)
                *
                *
                * BLOCK 6.   CERR=ERROR+CUM
                *            CERR=CERR/2  (RESCALE CERR)
                * (MOST SIGNIFICANT 8 BITS OF CUM IN ACC A)
                *
0150 BB 0406         ADD A  ERROR      ERROR+CUM=CERR
0153 47              ASR A  CERR/2
                *
                *
                * BLOCK 7.   CERR=CLAMP(CERR)
                *
0154 C6 07           LDA B  #7         CLAMP FOR CERR (=.5 VOLTS)
0156 BD 018F         JSR    CLAMP
0159 B7 040B         STA A  CERR       SAVE CERR (8 BITS)
                *
                *
                * BLOCK 8.   READ FSTAR    FSTAR=ABS(FSTAR)
                *
015C F6 040C         LDA B  FSTAR      FREQ REF
015F 2A 01           BPL    *+3
0161 50              NEG B  ABS        FSTAR
                *
                *
                * BLOCK 9.   VSTAR=CERR+FSTAR
                *            VSTAR=CLAMP(VSTAR)
                * (CERR IS IN ACC A, FSTAR IS IN ACC B)
                *
0162 1B              ABA               CERR+FSTAR
0163 2A 01           BPL    *+3
0165 4F              CLR A  IF         VSTAR NEG, VSTAR=0
0166 C6 4D           LDA B  #$4D       MAX VSTAR (=6V)
0168 11              CBA               CLAMP VSTAR TO +6V
0169 2B 01           BMI    *+3
016B 17              TBA
016C B7 040D         STA A  VSTAR      SAVE VSTAR
016F 7E 0100         JMP    START      REPEAT PROGRAM
```

```
                   *
                   * SMPY--SIGNED MULTIPLY ROUTINE
                   *
                   *
                   * CALLING SEQUENCE:  JSR     SMPY
                   * INPUTS:  ACCA=MULTIPLICAND
                   *          ACCB=MULTIPLIER
                   * OUTPUTS: 16 BIT PRODUCT IN ACCA,ACCB
                   *
0172 B7 040E SMPY   STA A  MCAND    SAVE MULTIPLICAND
0175 4F             CLR A  INIT     RESULT
0176 CE 0008        LDX    #8       8 BITS
                   *
0179 C5 01   SMPY1  BIT B  #1       TEST LSB OF MULTIPLIER
017B 27 05          BEQ    SMPY2    IF ZERO, NO ADD
017D BB 040E        ADD A  MCAND    ADD MULTIPLICAND TO PARTIAL
0180 29 02          BVS    SMPY3    TEST FOR ARITH OVERFLOW
0182 47      SMPY2  ASR A  EXTEND   SIGN
0183 49             ROL A
0184 46      SMPY3  ROR A  POSITION PARTIAL RESULT
0185 56             ROR B
0186 09             DEX             COUNT BITS
0187 26 F0          BNE    SMPY1    GO BACK IF MORE BITS
                   *
0189 24 03          BCC    SMPY4    TEST SIGN OF MULTIPLIER
018B B0 040E        SUB A  MCAND    IF NEG, ADJUST RESULT
018E 39      SMPY4  RTS             RETURN
```

```
                    *
                    * CLAMP--SUBROUTINE TO CLAMP ACC A TO +/- ACC B
                    *
                    * INPUTS:  ACCA=VALUE TO BE CLAMPED
                    *          ACCB=CLAMP VALUE (MUST BE POSITIVE)
                    *
                    * OUTPUTS: IF(ACCA > ACCB) THEN ACCA=ACCB
                    *          IF(ACCA < -ACCB) THEN ACCA=-ACCB
                    *          ELSE ACCA IS UNCHANGED
                    *
018F 8A 00          CLAMP  ORA A  #0        TEST SIGN OF A
0191 2B 05                 BMI    CLMP1     BRANCH IF NEG
                    *
0193 11                    CBA              COMPARE ACCUMULATORS
0194 2B 01                 BMI    *+3       IF MINUS A IS OK
0196 17                    TBA              ELSE SET ACCA=ACCB
0197 39                    RTS              RETURN
                    *
0198 50             CLMP1  NEG B  NEGATE    CLAMP
0199 11                    CBA              COMPARE ACCUMULATORS
019A 2A 01                 BPL    *+3       IF PLUS A IS OK
019C 17                    TBA              ELSE SET ACCA=-ACCB
019D 50                    NEG B  RESTORE   ACCB
019E 39                    RTS              RETURN
```

```
                   *
                   * DEFINE DATA LOCATIONS
                   *
      0400                ORG    $400
      0400 00      UD     FCB    0        INPUTS (8 BIT VARIABLES)
      0401 00      ID     FCB    0
      0402 00      UQ     FCB    0
      0403 00      IQ     FCB    0
      0404 0000    UID    FDB    0        UD*ID (16 BITS)
      0406 00      ERROR  FCB    0        ERROR SIGNAL
      0407 0000    ER256  FDB    0        ERROR/256 (16 BITS)
      0409 0000    CUM    FDB    0        CUMULATIVE ERROR (16 BITS)
      040B 00      CERR   FCB    0        COMPENSATED ERROR
      040C 00      FSTAR  FCB    0        FREQ REF
      040D 00      VSTAR  FCB    0        VOLTAGE REQUEST
      040E 00      MCAND  FCB    0        MULTIPLICAND STORAGE
                   *
                          END

TOTAL ERRORS 00000
```

APPENDIX B

BASIC ASPECTS OF THE MODULATION MECHANISM

The type of pulse width modulation studied applies to inverter systems made up of three thyristor poles as shown on Figure 1, forming a "3-phase bridge". The poles are connected across a dc link E_d, consisting of a dc source of fixed magnitude. It is practical to define the pole output voltages V_{AO}, V_{BO} and V_{CO}, relative to the mid-point voltage 0 of the dc source, here termed dc neutral. Therefore, the dc source is represented by two batteries of $E_d/2$ volts each, although in an actual system the dc neutral may not be physically accessible. Each transistor has a free-wheeling diode in antiparallel connection and the firing sequence is such that as one transistor in a given pole is turned on, the reciprocal one in the same pole goes off. Therefore, the transistor system of Figure 1 can be represented by a bi-directional, double throw switch system The switches are periodically and independently activated from one position to the other, connecting alternately the positive and negative terminal of the dc source to the load buses. The object of the modulation strategy is to perform the switching with such timing and sequence that at the terminals of the load a 3-phase ac voltage appears whose fundamental component has the desired amplitude and frequency, and whose other components have a minimum of unwanted side effects. In addition to the pole-to-neutral instantaneous voltage V_{AO}, V_{BO} and V_{CO} ("pole voltages"), the instantaneous pole-to-pole voltages V_{AB} V_{BC} and V_{CA} ("line voltages") can also be considered.

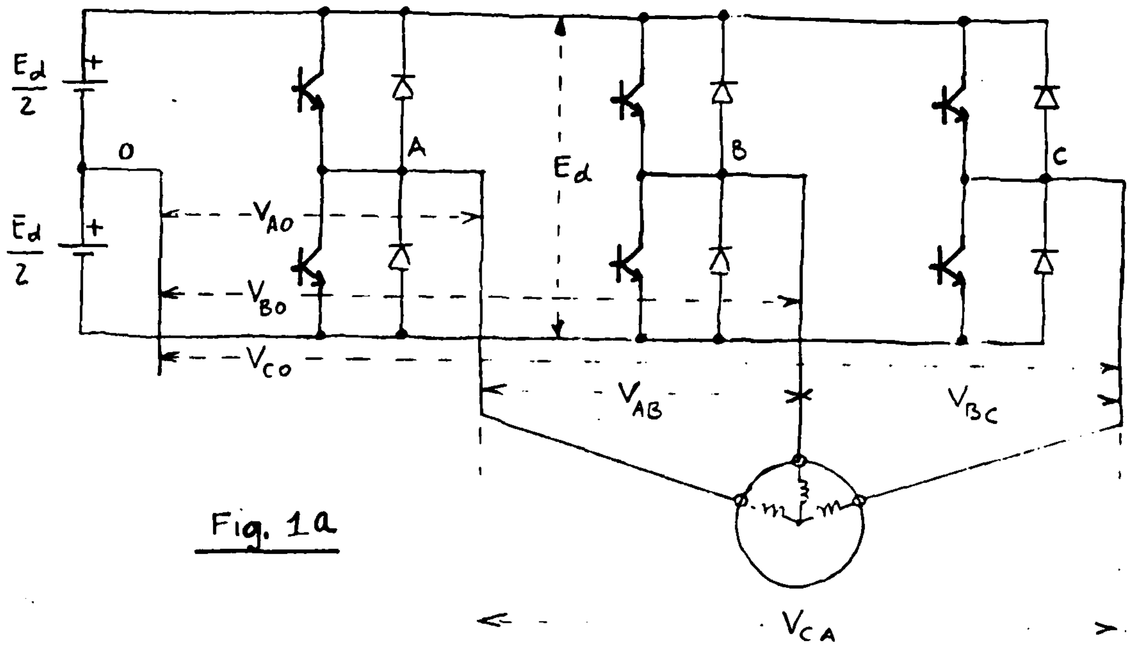


Fig. 1a

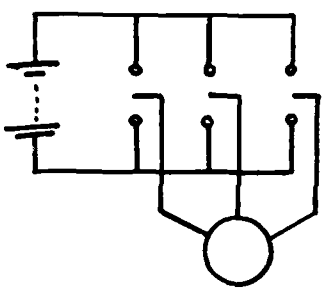


Fig. 1b

$$|E_{AB}| = |E_{BC}| + |E_{CA}| \quad \text{at any instant}$$

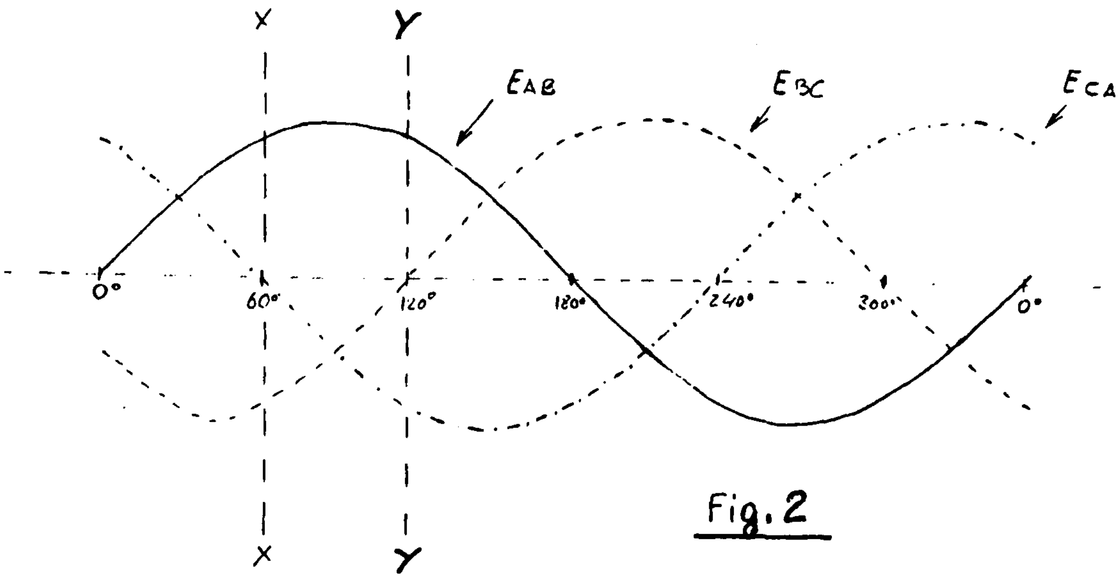


Fig. 2

Permissible Levels of the Line Voltages

It can be seen that a given pole voltage can assume only two discrete levels, $+E_d/2$ or $-E_d/2$, or for brevity, + and -. On the other hand, a line voltage can assume three discrete levels, a + level corresponding to $+E_d$ volts, a - level corresponding to $-E_d$ volts and a 0 level, obtained when the two corresponding poles are in the same state.

While there is no limitation in our freedom to impose to each one of the three pole voltages either of the two possible levels we desire the situation is different as far as the line voltages are concerned. In this case, imposing a given level to one of the line voltages interacts with the freedom of the others to assume any level. This is the inter phase compatibility constraint, better illustrated by considering the following table, listing on the left side all the possible combinations of levels that one can confer to the pole voltages (8 in all, one per row) and on the right side the corresponding combinations of line voltage levels that result from the imposed switch configurations.

V_{AO}	V_{BO}	V_{CO}	V_{AB}	V_{BC}	V_{CA}
+	+	+	0	0	0
-	+	+	-	0	+
+	-	+	+	-	0
+	+	-	0	+	-
-	-	+	0	-	+
+	-	-	+	0	-
-	+	-	-	+	0
-	-	-	0	0	0

Whenever a line voltage has one of the three possible levels (+, - or 0), the other two line voltages must have the other two possible levels, excluding the possibility for two or three line voltages to share the same level. An exception to the rule is the "all zero" combination of line voltages, for which there are two pole combinations.

Hexamerous Symmetry of the Output Cycle

Considering the fundamental component of each of the three line voltages (sine waves E_{AB}, E_{BC} and E_{CA}, Figure 2), **one entire** output cycle can be divided in six identical segments, such as the one comprised between the vertical axes XX and YY. In each segment, one of the line sine waves is peculiar in that it rides through its crest with a given polarity, whereas the others rise from zero or decline to zero over the segment with opposite polarity. We shall call "dominant" line voltage in any given segment the line voltage whose fundamental sine wave rides through its crest in the considered segment. The other two line voltages shall be called "complementary" line voltages. The "first complementary" line voltage is the one whose fundamental rises from zero in the considered segment. The "second complementary" line voltage is the one whose fundamental declines to zero in the considered segment. In the segment between the XX and YY axes, Figure 2, E_{AB} is the fundamental of the dominant line, and E_{CA} is the fundamental of the 1st complementary line. Because the segments repeat themselves each 60° interval, with identical relationships between dominant and complementary lines, it suffices to consider a single segment to study the modulation requirements. The relationships of interest are as follows: dominant and complementary

lines always have opposite polarity, except at the segment borderline. The dominant always exceeds either complementary in absolute value. Both complementary lines have same polarity. At any instant, the absolute values of the two complementary lines add up to the absolute value of the dominant.

Since, at any given time the three line voltages are riding through one of the segments of the output cycle, one of them is dominant at that time and the other two are complementary. The modulation mechanism must be capable of manipulating one of the line voltages through a 60° interval according to rules that confer to it the role of dominant line, and at the same time it must be manipulating the other two as required to confer to them the proper complementary role.

Pulse Polarity Consistency Rule

Even without entering into the details of the waveform harmonic analysis, one can state a sensible ground rule that the modulation strategy should follow. That is, the pulses which form the pole-to-pole waveform should have a consistent polarity throughout one half cycle of the fundamental sine-wave. This means that the components of the string of pulses which make up the positive (or negative) half wave can be of variable duration or variable spacing to suit the modulation requirements, but should all be positive (or negative). Figures 3a and 3b illustrate the concept. Both waveforms were carefully constructed to have the same fundamental amplitude, frequency and phase, and the same number of pulses, but Figure 3a conforms to the polarity consistency rule and Figure 3b does not conform. The non-consistent pulses of Figure 3b have

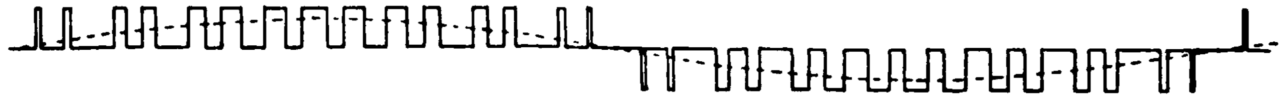


Fig. 3a

Fig. 3b

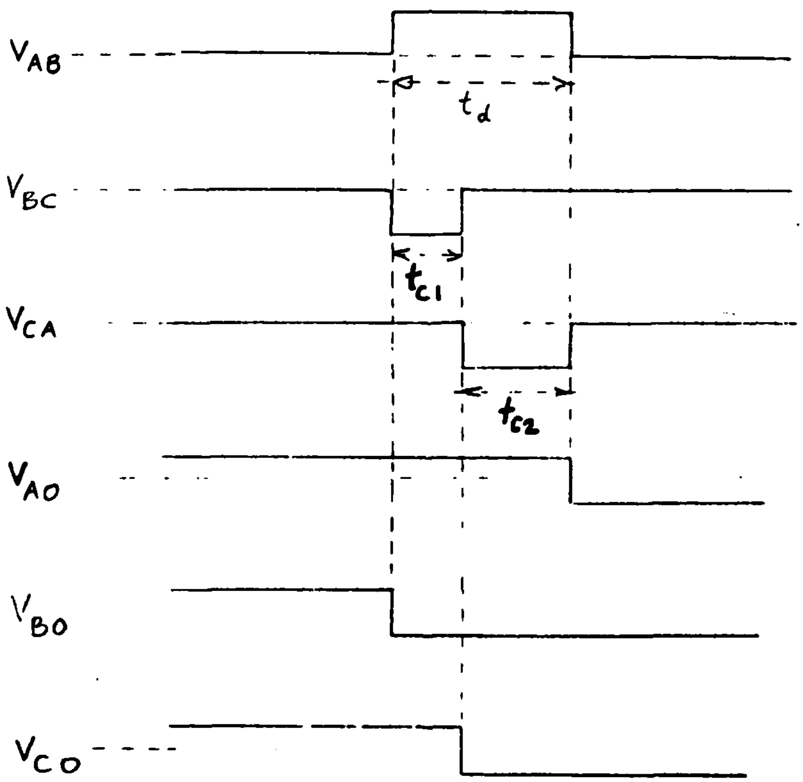


Fig. 4

no effect on the fundamental, being offset by corresponding width variations of the consistent pulses. However, they adversely affect the high order harmonic spectrum and should be avoided by using a proper modulating mechanism.

Optimal Pole Switching Sequence

The above rules and principles can be applied to find out in which order the three poles should typically change state. Assume that at a given instant line voltage V_{AB} is dominant and its fundamental is positive. V_{AB} will, therefore, be made up of a series of positive pulses, one of which is represented on Figure 4 with a width t_d.

a. During time t_d, the two complementary lines can only be negative or zero, as shown in the table of permissible levels. One could be negative for the t_d duration and the other zero for the same duration. However, to play their complementary role, they should both be negative on the average during t_d. Therefore, one of them, V_{BC} for instance, should become negative at the beginning of the t_d duration, and stay in that state for a time t_{C1}, after which V_{BC} returns to the zero level and V_{CA} becomes negative for a time t_{C2} (Figure 4), such that $t_{C1} + t_{C2} = t_d$. The ratio of times t_{C1} and t_{C2} should be the same as the ratio between the two complementary lines' fundamental voltages at the considered instant, and is, therefore, a function of the angular position of the dominant pulse within the output cycle segment.

b. Before and after time t_d, when the dominant line is zero, the two complementaries must also be zero. For the only other permissible combination of levels would be that one complementary is positive and the other negative, but this violates the pulse polarity consistency rule.

c. Before the initiation of a dominant pulse, the all-zero combination of line levels must be achieved with one of the following pole state combinations:

A = (+) B = (+) C = (+) (1)

A = (-) B = (-) C = (-) (2)

Assuming state (1), to generate the dominant pulse, pole B must switch from B = (+) to B = (-) at the beginning of the time t_d (see Figure 4) and then A and B must stay in their respective states for the duration of t_d. At some instant during t_d, determined by the t_{C1}/t_{C2} ratio requirement, pole C must follow the transition of B, i.e., must switch from C = (+) to C = (-). At the end of time t_d, all poles must assume the same state, and the most economical way of accomplishing that, in terms of total number of switchings, is to have pole A following the transitions of B and C, i.e., switching from A = (+) to A = (-).

d. The above reasoning shows that from a situation such as

A = (+) B = (+) C = (+) (1)

a dominant positive pulse is generated by changing the state of the poles in the sequence (B-C-A), ending up with the following situation:

A = (-) B = (-) C = (-) (2)

Similarly from situation (2), the next dominant positive pulse can be generated by changing the state of the pulses in the sequence (A-C-B) ending up with the original situation (1).

The optimal pole commutation sequence is, therefore, B-C-A, A-C-B, B-C-A, A-C-B, etc., resulting in the "nested" or "encased" pole-pole

waveform configuration of Figure 5. The sequence should continue unchanged throughout an output cycle 60° segment. At the boundary between segments, after a permutation between the role of the poles, accounting for the fact that another line becomes dominant, a similar sequence takes effect.

The following table therefore applies:

Dominant Line Voltage in the Considered Segment	Polarity of the Dominant Fundamental in the Considered Segment	Optimum Pole Commutation Sequence
V_{AB}	+	B↘ , C↘ , A↘ , A↗ , C↗ , B
	−	A↘ , C↘ , B↘ , B↗ , C↗ , A↗
V_{BC}	+	C↘ , A↘ , B↘ , B↗ , A↗ , C
	−	B↘ , A↘ , C↘ , C↗ , A↗ , B↗
V_{CA}	+	A↘ , B↘ , C↘ , C↗ , B↗ , A
	−	C↘ , B↘ , A↘ , A↗ , B↗ , C↗

B↘ means: pole voltage V_{B0} commutates from the (+) level to the (-) level

A↗ means: pole voltage V_{A0} commutates from the (-) level to the (+) level

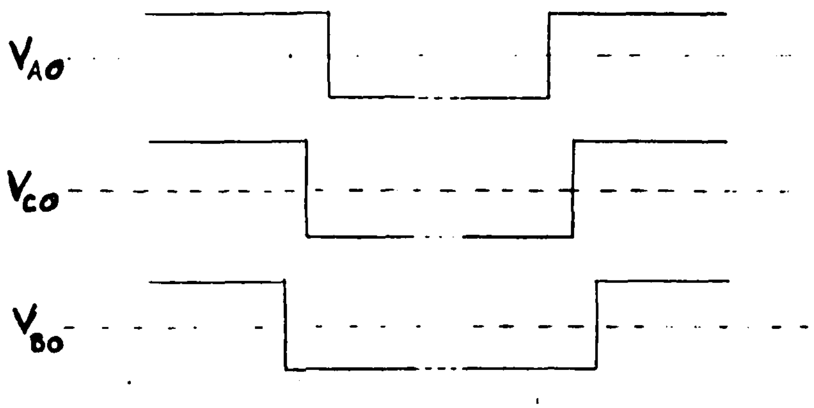


Fig. 5

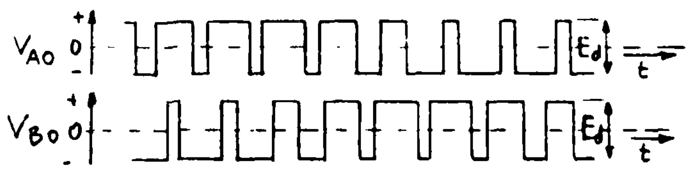


Fig 6a

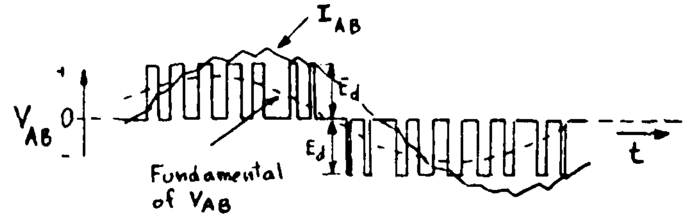


Fig 6b

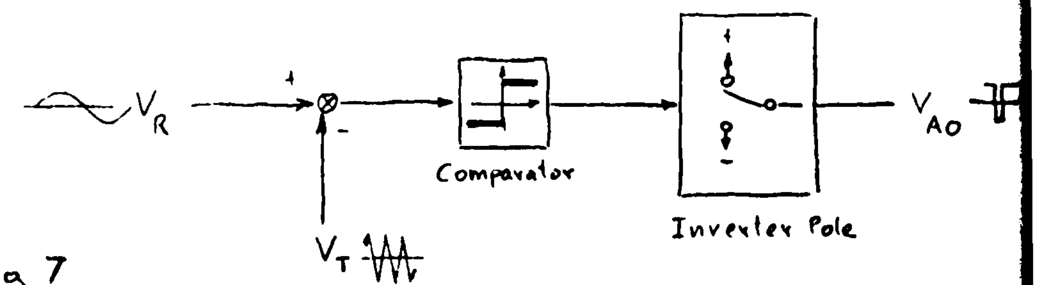


Fig 7

Assume a 60^{o} interval such as that shown in Figure 2. Generation of the output voltage of the dominant phase will require a series of pulses. The more pulses that can be used, the closer the output current can be made to approach a sine wave. Divide the 60^{o} interval into an arbitrary number of pulse intervals t_{pi}. There will be a pulse of length $t_{d1 \longrightarrow n}$ (where t_d varies across the 60^{o} interval) in each pulse internal t_{pi} and also the complementary pulses t_{c1}, t_{c2} for the other phases. The dominant pulse length t_d will vary such that $E_d \times t_d$ in any pulse internal t_{pi} will equal the volt time interval ($\int Vdt$) of the desired sine wave during the corresponding t_{pi} interval.

The pulse width is then a sinusoidal function of time. Similarly t_{c1}, and t_{c2} are sinusoidal functions of time across the 60^{o} interval. To vary the amplitude of the desired output, the pulse width of all the pulses is varied by a constant multiple, thus preserving the sinusoidal relationship across the 60^{o} interval. The maximum amplitude, while still generating a 'sine wave' of current in the output, is reached when one t_d pulse fills the whole interval and 'saturation' is reached.

It is now possible to develop techniques for generating pulse patterns that implement the generalized PWM strategy. The traditional analog technique which produces waveforms that fit the rules for optimum modulation strategy uses a triangular carrier waveform which is compared with 3 phase reference waveforms and switches each pole - or + depending on whether the triangle is greater than the reference for that phase or smaller. At one fixed frequency a fixed number of carrier cycles can be fitted inside a reference waveform half cycle. For variable frequency systems unsymmetric pulse patterns (with the effect of low frequency components in the output) arise as non-integral numbers of carrier cycles fit inside a reference half cycle.

The use of digital techniques permits an approach which guarantees a proper number of t_{pi} intervals in each 60^{o} interval and avoids the low frequency beat problem. The steps for generating the waveforms for a digital approach are outlined below.

- Set desired output frequency.
- Look up (in Read Only Memory) optimum number of t_{pi} for this frequency.
- Look up (in ROM) base value for t_d, t_{c1} t_{c2} for each t_{pi}.
- Input desired output voltage.
- Multiply td, t_{c1}, t_{c2} pulses with voltage factor.
- Decide which phase is dominant this 60^{o} interval. Output pulse train to power stage.

Estimating the Pole Firing Times

A Fortran program was written for a Data General Nova Computer to compute the switching times for each of the three poles. A triangle wave was simulated and each of the three phases of the sine wave was compared to it. A summary print out displayed the crossover times for the signals as well as their approximate values, switching sequence and pole and line voltages. See Figures 8 – 12 for a sample output.

APPENDIX C

HARMONIC ANALYSIS PROGRAM

The program appears on the following page. Note that the outputs actually published in this appendix do not print "n DB threshold for printout" and do "m integration steps per increment". Those differences from the program included herein exist because the printout shown is taken from an earlier version of the harmonic analysis program which is substantially the same as the one published. A sample of the output of the latest program is shown below.

```
HARMONIC ANALYSIS OF SYNTHESIZED SINE WAVE USING
CONSTANT AMPLITUDE PWM WAVEFORM, AS IN DEVELOPMENT MOTOR CONTROLLER.

 9     PWM INCREMENTS PER OUTPUT CYCLE
 60    HZ OUTPUT FREQUENCY
-40    DB THRESHOLD FOR PRINTOUT
       PWM INCREMENTS CENTERED

COMPONENT        ABS AMPLITUDE   REL AMPLITUDE   REL AMPL IN DB

 60   HZ          0.758347572     1               0
 180  HZ          0.031593768     0.041661330    -27.60533744
 360  HZ          0.103020575     0.135840757    -17.33808661
 480  HZ          0.242079212     0.319219341    -9.918216057
 600  HZ          0.076857195     0.101348245    -19.88367535
 720  HZ          0.162526080     0.214316082    -13.37890480
 780  HZ          0.064951990     0.085649367    -21.34551684
 840  HZ          0.074941684     0.098822344    -20.10289702
 900  HZ          0.102216315     0.134788214    -17.40696161
```

```
10 PRINT "HARMONIC ANALYSIS OF SYNTHESIZED SINE WAVE USING"
20 PRINT "CONSTANT AMPLITUDE PWM WAVEFORM, AS IN DEVELOPMENT MOTOR CON.
30 PRINT
40 PRINT
50 DIM C[100],E[100],G[100],J[100],S[100]
   FOR M=1 TO 100
   C[M]=0
   E[M]=0
   G[M]=0
   J[M]=0
   S[M]=0
   NEXT M
   DISP "SPEED OUT, HIGHEST HARMONIC";
   INPUT S,H
   DISP "TYPE'1' IF INCREMENTS ARE CENTRD"
   WAIT 1000
   DISP "TYPE '0' OTHERWISE";
   WAIT 1000
   INPUT Q
   DISP "NO. OF INCREMENTS/OUTPUT CYCLE";
   INPUT N
   DISP "AMPL. THRESHOLD FOR PRINTOUT(DB)";
158 INPUT Y
159 Y1=10↑(Y/20)
   X1=2*PI/N
   PRINT N;"PWM INCREMENTS PER OUTPUT CYCLE"
   PRINT S;"HZ OUTPUT FREQUENCY"
   PRINT Y;"DB THRESHOLD FOR PRINTOUT"
   IF Q=0 THEN 200
   PRINT "      PWM INCREMENTS CENTERED"
   GOTO 210
   PRINT "      PWM INCREMENTS LEFT SIDE JUSTIFIED"
   PRINT
   PRINT
   PRINT "COMPONENT","ABS AMPLITUDE","REL AMPLITUDE","REL AMPL IN DB"
   PRINT
   FOR Z1=0 TO (2*PI-X1) STEP X1
   D=(COS(Z1)-COS(Z1+X1))/X1
   FOR M=1 TO H
344 IF Q=0 THEN 349
345 Z8=Z1+(1-D)*X1/2
   GOTO 350
   Z8=Z1
   C[M]=COS(Z8*M)/M-COS(M*D*X1+M*Z8)/M
   G[M]=SIN(M*D*X1+M*Z8)/M-SIN(Z8*M)/M
   J[M]=J[M]+G[M]
   E[M]=E[M]+C[M]
   NEXT M
   NEXT Z1
   FOR M=1 TO H
   S[M]=SQR(E[M]↑2+J[M]↑2)
   IF S[M]/S[1]<Y1 THEN 440
   PRINT M*S;"HZ",S[M]/4,S[M]/S[1],20*LGT(S[M]/S[1])
   NEXT M
   PRINT "------
   END
```

75 PWM INCREMENTS PER OUTPUT CYCLE
5 INTEGRATION STEPS PER INCREMENT
7 HZ OUTPUT FREQUENCY
PWM INCREMENTS CENTERED

COMPONENT	ABS AMPLITUDE	P.C. REL FUND	DB REL FUND
7	.784978	100	0
490	2.09044E-2	2.66306	-31.4924
504	.162348	20.6818	-13.6882
518	.152733	19.4569	-14.2185
532	.132432	16.8707	-15.4573
546	.16999	21.6554	-13.2887
560	3.17172E-2	4.04052	-27.8713
595	2.17304E-2	2.76829	-31.1558
609	.163155	20.7847	-13.6451
623	.150953	19.2303	-14.3203
637	.134044	17.0761	-15.3522
651	.169526	21.5962	-13.3124
665	3.07466E-2	3.91687	-28.1412
1001	3.36916E-2	4.29205	-27.3467
1015	9.45968E-2	12.0509	-18.3796
1029	1.72731E-2	2.20045	-33.1498
1043	5.62196E-2	7.16193	-22.8994

---

66 PWM INCREMENTS PER OUTPUT CYCLE
5 INTEGRATION STEPS PER INCREMENT
8 HZ OUTPUT FREQUENCY
PWM INCREMENTS CENTERED

COMPONENT	ABS AMPLITUDE	P.C. REL FUND	DB REL FUND
8	.784856	100	0
488	2.02416E-2	2.57902	-31.7709
504	.161665	20.5981	-13.7235
520	.1542	19.6469	-14.1341
536	.131124	16.7067	-15.5422
552	.170344	21.7038	-13.2693
560	2.09113E-2	2.66435	-31.4882
568	3.25119E-2	4.1424	-27.655
576	.162352	20.6856	-13.6867
592	.152735	19.4602	-14.2171
608	.132431	16.8732	-15.456
624	.169989	21.6587	-13.2874
640	3.17167E-2	4.04109	-27.87
680	2.17306E-2	2.76874	-31.1544
696	.163155	20.7879	-13.6438
712	.150954	19.2333	-14.3189
728	.134044	17.0788	-15.3509
744	.169526	21.5996	-13.3111
760	3.07466E-2	3.91748	-28.1399
1000	.032916	4.19389	-27.5477
1016	9.46747E-2	12.0627	-18.3711
1032	1.86955E-2	2.38203	-32.461
1048	5.66596E-2	7.21911	-22.8303
1064	4.98152E-2	6.34705	-23.9486
1096	8.99028E-2	11.4547	-18.8203
1112	4.52183E-2	5.76135	-24.7895
1144	3.21975E-2	4.10234	-27.7394
1160	9.44492E-2	12.034	-18.3918
1176	1.72628E-2	2. ◆INTERRUPTED◆	

HARMONIC ANALYSIS OF SYNTHESIZED SINE WAVE USING
CONSTANT AMPLITUDE PWM WAVEFORM.

60 PWM INCREMENTS PER OUTPUT CYCLE
5 INTEGRATION STEPS PER INCREMENT
9 HZ OUTPUT FREQUENCY
PWM INCREMENTS CENTERED

COMPONENT	ABS AMPLITUDE	P.C. REL FUND	DB REL FUND
9	.784742	100	0
495	1.96967E-2	2.50996	-32.0067
513	.16108	20.5265	-13.7537
531	.155427	19.8061	-14.064
549	.130045	16.5718	-15.6126
567	.170629	21.7434	-13.2535
585	3.31944E-2	4.22998	-27.4732

---

54 PWM INCREMENTS PER OUTPUT CYCLE
5 INTEGRATION STEPS PER INCREMENT
10 HZ OUTPUT FREQUENCY
PWM INCREMENTS CENTERED

COMPONENT	ABS AMPLITUDE	P.C. REL FUND	DB REL FUND
10	.784588	100	0
490	1.90428E-2	2.42711	-32.2982
510	.160366	20.4395	-13.7906
530	.157808	20.1135	-13.9302
550	.109039	13.8976	-17.1412
570	9.87375E-3	1.25846	-38.0032
590	.121404	15.4736	-16.2082
610	.133283	16.9877	-15.3973
630	.170823	21.7723	-13.2419
650	3.32023E-2	4.23181	-27.4695

---

48 PWM INCREMENTS PER OUTPUT CYCLE
5 INTEGRATION STEPS PER INCREMENT
11 HZ OUTPUT FREQUENCY
PWM INCREMENTS CENTERED

COMPONENT	ABS AMPLITUDE	P.C. REL FUND	DB REL FUND
11	.784373	100	0
473	1.82408E-2	2.32552	-32.6696
495	.159418	20.3242	-13.8397
517	.159642	20.3528	-13.8275
539	.108071	13.778	-17.2163
561	1.09629E-2	1.39766	-37.092
583	.12274	15.6482	-16.1107
605	.11252	14.3453	-16.8658
627	1.00941E-2	1.2869	-37.8091
649	.121394	15.4766	-16.2065
671	.133283	16.9923	-15.3949
693	.170823	21.7783	-13.2395
715	3.32022E-2	4.23295	-27.4671

---

45 PWM INCREMENTS PER OUTPUT CYCLE
5 INTEGRATION STEPS PER INCREMENT
12 HZ OUTPUT FREQUENCY
PWM INCREMENTS CENTERED

COMPONENT	ABS AMPLITUDE	P.C. REL FUND	DB REL FUND
12	.784232	100	0
480	1.77689E-2	2.26577	-32.8957
504	.158821	20.2518	-13.8707
516	1.82374E-2	2.32551	-32.6696
528	.159995	20.4015	-13.8068
540	.159402	20.3259	-13.839
552	.126151	16.0859	-15.8711
564	.159636	20.3558	-13.8263
576	.171551	21.875	-13.201
588	.108066	13.7799	-17.2151
600	3.56977E-2	4.55193	-26.8361
612	1.09602E-2	1.39757	-37.0925
636	.122741	15.6512	-16.1091
660	.11252	14.3477	-16.8643
684	1.00938E-2	1.28709	-37.8078
708	.121394	15.4794	-16.2049
732	.133283	16.9954	-15.3934
756	.170823	21.7822	-13.238
780	3.32022E-2	4.23372	-27.4656

---

39 PWM INCREMENTS PER OUTPUT CYCLE
5 INTEGRATION STEPS PER INCREMENT
13 HZ OUTPUT FREQUENCY
PWM INCREMENTS CENTERED

COMPONENT	ABS AMPLITUDE	P.C. REL FUND	DB REL FUND
13	.783847	100	0
442	1.66339E-2	2.12209	-33.4647
468	.157347	20.0738	-13.9474
494	.163516	20.8607	-13.6134
520	.106033	13.5272	-17.3758
546	1.31788E-2	1.6813	-35.4871
559	1.82412E-2	2.32714	-32.6635
572	.122731	15.6575	-16.1056
585	.159405	20.3362	-13.8346
598	.13018	16.6079	-15.5937
611	.159637	20.3659	-13.8219
624	.171836	21.9222	-13.1823
637	.108066	13.7866	-17.2109
650	3.57124E-2	4.55605	-26.8282
663	1.09599E-2	1.39822	-37.0885
689	.122742	15.6589	-16.1048
715	.11252	14.3548	-16.8601
741	1.00938E-2	1.28773	-37.8035
767	.121394	15.487	-16.2007
793	.133283	17.0037	-15.3891
819	.170823	21.793	-13.2337
845	3.32106E-2	4.23688	-27.4591

36 PWM INCREMENTS PER OUTPUT CYCLE
5 INTEGRATION STEPS PER INCREMENT
14 HZ OUTPUT FREQUENCY
PWM INCREMENTS CENTERED

COMPONENT	ABS AMPLITUDE	P.C. REL FUND	DB REL FUND
14	.783578	100	0
434	1.59462E-2	2.03505	-33.8285
462	.156375	19.9566	-13.9983
476	1.66315E-2	2.12251	-33.463
490	.164642	21.0115	-13.5508
504	.157335	20.0791	-13.9451
518	.122355	15.6149	-16.1292
532	.163511	20.8673	-13.6107
546	.172249	21.9824	-13.1585
560	.106029	13.5314	-17.3731
574	.037528	4.78932	-26.3945
588	1.31767E-2	1.68161	-35.4855
602	1.39502E-2	1.78032	-34.99
616	.122732	15.663	-16.1025
630	.159086	20.3025	-13.849
644	.13018	16.6135	-15.5908
658	.15962	20.3706	-13.8199
672	.171836	21.9297	-13.1794
686	.108066	13.7914	-17.2078
700	3.57123E-2	4.5576	-26.8253
714	1.09599E-2	1.3987	-37.0855
742	.122742	15.6642	-16.1018
770	.11252	14.3598	-16.8571
798	1.00943E-2	1.28823	-37.8001
826	.121387	15.4914	-16.1982

---

36 PWM INCREMENTS PER OUTPUT CYCLE
5 INTEGRATION STEPS PER INCREMENT
15 HZ OUTPUT FREQUENCY
PWM INCREMENTS CENTERED

COMPONENT	ABS AMPLITUDE	P.C. REL FUND	DB REL FUND
15	.783578	100	0
465	3.18908E-2	4.06989	-27.8083
495	.312748	39.9128	-7.97775
510	1.66291E-2	2.1222	-33.4643
525	.329278	42.0223	-7.5304
540	.157323	20.0775	-13.9458
555	.244709	31.2297	-10.1086
570	.163507	20.8667	-13.6109
585	.344501	43.9651	-7.13783
600	.106026	13.531	-17.3734
615	7.57787E-2	9.67086	-20.2907
630	1.31746E-2	1.68133	-35.4869
645	9.65923E-3	1.23271	-38.1828
660	.122733	15.6631	-16.1024
675	.158767	20.2619	-13.8664
690	.130179	16.6135	-15.5908
705	.159602	20.3684	-13.8209
720	.171836	21.9296	-13.1794
735	.108067	13.7915	-17.2078
750	3.57123E-2	4.55759	-26.8253
765	1.09599E-2	1.3987	-37.0855
795	.122741	15.6642	-16.1018
825	.11252	14.3598	-16.857

---

33 PWM INCREMENTS PER OUTPUT CYCLE
5 INTEGRATION STEPS PER INCREMENT
16 HZ OUTPUT FREQUENCY
PWM INCREMENTS CENTERED

COMPONENT	ABS AMPLITUDE	P.C. REL FUND	DB REL FUND
16	.783233	100	0
448	1.51523E-2	1.93458	-34.2683
480	.155202	19.8155	-14.0599
496	3.18886E-2	4.07141	-27.8051
512	.167362	21.3681	-13.4047
528	.312736	39.9289	-7.97424
544	.104034	13.2826	-17.5343
560	.329273	42.0403	-7.52668
576	1.51905E-2	1.93946	-34.2464
592	.244706	31.2431	-10.1049
608	.124083	15.8425	-16.0035
624	.344499	43.9843	-7.13406
640	.110641	14.1262	-16.9995
656	7.57779E-2	9.67501	-20.287
672	1.35366E-2	1.7283	-35.2476
688	9.65964E-3	1.2333	-38.1786
704	.122711	15.6673	-16.1001
720	.158768	20.2708	-13.8626
736	.13018	16.6209	-15.5869
752	.159602	20.3774	-13.817
768	.171836	21.9393	-13.1755
784	.108067	13.7976	-17.2039
800	3.57124E-2	4.55962	-26.8214
816	1.09603E-2	1.39936	-37.0814
848	.122736	15.6704	-16.0984

---

30 PWM INCREMENTS PER OUTPUT CYCLE
5 INTEGRATION STEPS PER INCREMENT
17 HZ OUTPUT FREQUENCY
PWM INCREMENTS CENTERED

COMPONENT	ABS AMPLITUDE	P.C. REL FUND	DB REL FUND
17	.782779	100	0
425	1.42317E-2	1.8181	-34.8076
459	.153748	19.6414	-14.1366
493	.169368	21.6368	-13.2961
527	8.67727E-2	11.0852	-19.1051
544	.167358	21.3799	-13.3999
561	.139962	17.8802	-14.9526
578	.104031	13.29	-17.5295
595	.288434	36.8474	-8.67185
612	1.51887E-2	1.94035	-34.2424
629	.249732	31.9032	-9.92332
646	.124084	15.8517	-15.9985
663	.344919	44.0634	-7.11843
680	.110641	14.1343	-16.9945
697	7.58048E-2	9.68406	-20.2788
714	1.35365E-2	1.72929	-35.2426
731	9.65820E-3	1.23383	-38.1749
748	.122711	15.6763	-16.0951
765	.158767	20.2825	-13.8576
782	.130181	16.6306	-15.5819
799	.159598	20.3886	-13.8122
816	.171836	21.9521	-13.1705
833	.108204	13.8231	-17.1879
850	3.57135E-2	4.56239	-26.8161

---

30 PWM INCREMENTS PER OUTPUT CYCLE
5 INTEGRATION STEPS PER INCREMENT
18 HZ OUTPUT FREQUENCY
PWM INCREMENTS CENTERED

COMPONENT	ABS AMPLITUDE	P.C. REL FUND	DB REL FUND
18	.782779	100	0
450	1.42317E-2	1.8181	-34.8076
486	.153748	19.6414	-14.1366
522	.169368	21.6368	-13.2961
558	.205434	26.2442	-11.6193
576	.167353	21.3794	-13.4001
594	3.28119E-2	4.19172	-27.5521
612	.104028	13.2896	-17.5298
630	.247595	31.6302	-9.99795
648	1.51869E-2	1.94013	-34.2434
666	.254757	32.5452	-9.75025
684	.124085	15.8518	-15.9984
702	.34534	44.1171	-7.10785
720	.11064	14.1343	-16.9945
738	7.58318E-2	9.6875	-20.2758
756	1.35364E-2	1.72928	-35.2427
774	9.65676E-3	1.23365	-38.1762
792	.122711	15.6763	-16.0951
810	.158767	20.2825	-13.8576
828	.130181	16.6306	-15.5818
846	.159594	20.3881	-13.8125

---

27 PWM INCREMENTS PER OUTPUT CYCLE
5 INTEGRATION STEPS PER INCREMENT
19 HZ OUTPUT FREQUENCY
PWM INCREMENTS CENTERED

COMPONENT	ABS AMPLITUDE	P.C. REL FUND	DB REL FUND
19	.782167	100	0
418	1.31518E-2	1.68146	-35.4863
456	.151902	19.4207	-14.2347
494	.172569	22.063	-13.1267
532	.116256	14.8634	-16.5576
570	.173017	22.1202	-13.1042
589	.205432	26.2644	-11.6126
608	.124774	15.9523	-15.9435
627	3.28104E-2	4.1948	-27.5458
646	.109589	14.0109	-17.0707
665	.247596	31.6551	-9.99113
684	1.56886E-2	2.00578	-33.9543
703	.254757	32.5707	-9.74347
722	.12405	15.8597	-15.9941
741	.34534	44.1517	-7.10106
760	.110643	14.1456	-16.9876
779	7.58347E-2	9.69547	-20.2686
798	1.35369E-2	1.7307	-35.2356
817	9.54795E-3	1.2207	-38.2678
836	.12271	15.6885	-16.0884

---

27 PWM INCREMENTS PER OUTPUT CYCLE
5 INTEGRATION STEPS PER INCREMENT
20 HZ OUTPUT FREQUENCY
PWM INCREMENTS CENTERED

COMPONENT	ABS AMPLITUDE	P.C. REL FUND	DB REL FUND
20	.782167	100	0
440	1.31518E-2	1.68146	-35.4863
480	.151902	19.4207	-14.2347
520	.172569	22.063	-13.1267
560	.116256	14.8634	-16.5576
600	.173017	22.1202	-13.1042
620	.205429	26.2641	-11.6127
640	8.21937E-2	10.5085	-19.5692
660	3.28088E-2	4.1946	-27.5462
680	.115149	14.7218	-16.6408
700	.247596	31.6552	-9.99111
720	1.61902E-2	2.06992	-33.6809
740	.254757	32.5706	-9.74347
760	.124014	15.8552	-15.9966
780	.34534	44.1517	-7.10105
800	.110645	14.1459	-16.9874
820	7.58377E-2	9.69585	-20.2683
840	1.35374E-2	1.73076	-35.2353

---

24 PWM INCREMENTS PER OUTPUT CYCLE
5 INTEGRATION STEPS PER INCREMENT
21 HZ OUTPUT FREQUENCY
PWM INCREMENTS CENTERED

COMPONENT	ABS AMPLITUDE	P.C. REL FUND	DB REL FUND
21	.781313	100	0
399	1.18724E-2	1.51954	-36.3658
441	.14949	19.1332	-14.3643
483	.176621	22.6056	-12.9157
525	.113315	14.5031	-16.7708
567	.173204	22.1683	-13.0854
609	4.47667E-2	5.72968	-24.8374
651	.211709	27.0966	-11.3417
672	8.21942E-2	10.52	-19.5597
693	3.34292E-2	4.27859	-27.374
714	.115149	14.7379	-16.6313
735	.247546	31.6833	-9.98339
756	1.61904E-2	2.07221	-33.6713
777	.254841	32.617	-9.73112
798	.124014	15.8725	-15.9871
819	.347286	44.449	-7.04275
840	.110646	14.1616	-16.9778

---

24 PWM INCREMENTS PER OUTPUT CYCLE
5 INTEGRATION STEPS PER INCREMENT
22 HZ OUTPUT FREQUENCY
PWM INCREMENTS CENTERED

COMPONENT	ABS AMPLITUDE	P.C. REL FUND	DB REL FUND
22	.781313	100	0
418	1.18724E-2	1.51954	-36.3658
462	.14949	19.1332	-14.3643
506	.176621	22.6056	-12.9157
550	.113315	14.5031	-16.7708
594	.173204	22.1683	-13.0854
638	4.47667E-2	5.72968	-24.8374
682	.217989	27.9004	-11.0878
704	8.21947E-2	10.5201	-19.5596
726	3.40495E-2	4.35799	-27.2143
748	.115149	14.7378	-16.6313
770	.247495	31.6769	-9.98516
792	1.61907E-2	2.07224	-33.6712
814	.254925	32.6278	-9.72825
836	.124013	15.8724	-15.9872

---

21 PWM INCREMENTS PER OUTPUT CYCLE
5 INTEGRATION STEPS PER INCREMENT
23 HZ OUTPUT FREQUENCY
PWM INCREMENTS CENTERED

COMPONENT	ABS AMPLITUDE	P.C. REL FUND	DB REL FUND
23	.78007	100	0
368	1.03427E-2	1.32587	-37.55
414	.146222	18.7447	-14.5424
460	.18191	23.3197	-12.6456
506	.109639	14.055	-17.0434
552	.173256	22.2104	-13.0689
598	4.75861E-2	6.10024	-24.2931
713	.218043	27.9518	-11.0718
736	8.21226E-2	10.5276	-19.5534
759	3.55942E-2	4.56296	-26.8151
782	.115156	14.7622	-16.617
805	.22757	29.1731	-10.7004
828	1.61932E-2	2.07587	-33.656

---

21 PWM INCREMENTS PER OUTPUT CYCLE
5 INTEGRATION STEPS PER INCREMENT
24 HZ OUTPUT FREQUENCY
PWM INCREMENTS CENTERED

COMPONENT	ABS AMPLITUDE	P.C. REL FUND	DB REL FUND
24	.78007	100	0
384	1.03427E-2	1.32587	-37.55
432	.146222	18.7447	-14.5424
480	.18191	23.3197	-12.6456
528	.109639	14.055	-17.0434
576	.173256	22.2104	-13.0689
624	4.75861E-2	6.10024	-24.2931
744	.218097	27.9587	-11.0697
768	8.20505E-2	10.5184	-19.561
792	.037139	4.76098	-26.4461
816	.115162	14.7631	-16.6165
840	.207645	26.6188	11.4962

21 PWM INCREMENTS PER OUTPUT CYCLE
5 INTEGRATION STEPS PER INCREMENT
25 HZ OUTPUT FREQUENCY
PWM INCREMENTS CENTERED

COMPONENT	ABS AMPLITUDE	P.C. REL FUND	DB REL FUND
25	.78007	100	0
400	1.03427E-2	1.32587	-37.55
450	.146222	18.7447	-14.5424
500	.18191	23.3197	-12.6456
550	.109639	14.055	-17.0434
600	.173256	22.2104	-13.0689
650	4.75861E-2	6.10024	-24.2931
775	.218151	27.9656	-11.0675
800	8.19784E-2	10.5091	-19.5687
825	3.86837E-2	4.959	-26.0921
850	.115169	14.764	-16.6159

18 PWM INCREMENTS PER OUTPUT CYCLE
5 INTEGRATION STEPS PER INCREMENT
26 HZ OUTPUT FREQUENCY
PWM INCREMENTS CENTERED

COMPONENT	ABS AMPLITUDE	P.C. REL FUND	DB REL FUND
26	.778161	100	0
78	8.67668E-3	1.11502	-39.0543
338	8.50392E-3	1.09282	-39.229
390	.141589	18.1953	-14.8008
442	.189096	24.3003	-12.2878
494	.104923	13.4834	-17.404
546	.173007	22.2329	-13.0601
598	5.13361E-2	6.59711	-23.6129
650	8.82124E-3	1.1336	-38.9108
754	1.72795E-2	2.22055	-33.0708
806	.307558	39.5237	-8.06285
832	8.19807E-2	10.5352	-19.5472

18 PWM INCREMENTS PER OUTPUT CYCLE
5 INTEGRATION STEPS PER INCREMENT
27 HZ OUTPUT FREQUENCY
PWM INCREMENTS CENTERED

COMPONENT	ABS AMPLITUDE	P.C. REL FUND	DB REL FUND
27	.778161	100	0
81	8.67668E-3	1.11502	-39.0543
351	8.50392E-3	1.09282	-39.229
405	.141589	18.1953	-14.8008
459	.189096	24.3003	-12.2878
513	.104923	13.4834	-17.404
567	.173007	22.2329	-13.0601
621	5.13361E-2	6.59711	-23.6129
675	8.82124E-3	1.1336	-38.9108
783	1.72795E-2	2.22055	-33.0708
837	.396965	51.0132	-5.84636

---

18 PWM INCREMENTS PER OUTPUT CYCLE
5 INTEGRATION STEPS PER INCREMENT
28 HZ OUTPUT FREQUENCY
PWM INCREMENTS CENTERED

COMPONENT	ABS AMPLITUDE	P.C. REL FUND	DB REL FUND
28	.778161	100	0
84	8.67668E-3	1.11502	-39.0543
364	8.50392E-3	1.09282	-39.229
420	.141589	18.1953	-14.8008
476	.189096	24.3003	-12.2878
532	.104923	13.4834	-17.404
588	.173007	22.2329	-13.0601
644	5.13361E-2	6.59711	-23.6129
700	8.82124E-3	1.1336	-38.9108
812	1.72795E-2	2.22055	-33.0708

---

18 PWM INCREMENTS PER OUTPUT CYCLE
5 INTEGRATION STEPS PER INCREMENT
29 HZ OUTPUT FREQUENCY
PWM INCREMENTS CENTERED

COMPONENT	ABS AMPLITUDE	P.C. REL FUND	DB REL FUND
29	.778161	100	0
87	8.67668E-3	1.11502	-39.0543
377	8.50392E-3	1.09282	-39.229
435	.141589	18.1953	-14.8008
493	.189096	24.3003	-12.2878
551	.104923	13.4834	-17.404
609	.173007	22.2329	-13.0601
667	5.13361E-2	6.59711	-23.6129
725	8.82124E-3	1.1336	-38.9108
841	1.72795E-2	2.22055	-33.0708

---

18 PWM INCREMENTS PER OUTPUT CYCLE
5 INTEGRATION STEPS PER INCREMENT
28 HZ OUTPUT FREQUENCY
PWM INCREMENTS CENTERED

COMPONENT	ABS AMPLITUDE	P.C. REL FUND	DB REL FUND
28	.778161	100	0
84	8.67668E-3	1.11502	-39.0543
364	8.50392E-3	1.09282	-39.229
420	.141589	18.1953	-14.8008
476	.189096	24.3003	-12.2878
532	.104923	13.4834	-17.404
588	.173007	22.2329	-13.0601
644	5.13361E-2	6.59711	-23.6129
700	8.82124E-3	1.1336	-38.9108
812	1.72795E-2	2.22055	-33.0708

18 PWM INCREMENTS PER OUTPUT CYCLE
5 INTEGRATION STEPS PER INCREMENT
29 HZ OUTPUT FREQUENCY
PWM INCREMENTS CENTERED

COMPONENT	ABS AMPLITUDE	P.C. REL FUND	DB REL FUND
29	.778161	100	0
87	8.67668E-3	1.11502	-39.0543
377	8.50392E-3	1.09282	-39.229
435	.141589	18.1953	-14.8008
493	.189096	24.3003	-12.2878
551	.104923	13.4834	-17.404
609	.173007	22.2329	-13.0601
667	5.13361E-2	6.59711	-23.6129
725	8.82124E-3	1.1336	-38.9108
841	1.72795E-2	2.22055	-33.0708

18 PWM INCREMENTS PER OUTPUT CYCLE
5 INTEGRATION STEPS PER INCREMENT
30 HZ OUTPUT FREQUENCY
PWM INCREMENTS CENTERED

COMPONENT	ABS AMPLITUDE	P.C. REL FUND	DB REL FUND
30	.778161	100	0
90	8.67668E-3	1.11502	-39.0543
390	8.50392E-3	1.09282	-39.229
450	.141589	18.1953	-14.8008
510	.189096	24.3003	-12.2878
570	.104923	13.4834	-17.404
630	.173007	22.2329	-13.0601
690	5.13361E-2	6.59711	-23.6129
750	8.82124E-3	1.1336	-38.9108
870	1.72795E-2	2.22055	-33.0708

15 PWM INCREMENTS PER OUTPUT CYCLE
5 INTEGRATION STEPS PER INCREMENT
31 HZ OUTPUT FREQUENCY
PWM INCREMENTS CENTERED

COMPONENT	ABS AMPLITUDE	P.C. REL FUND	DB REL FUND
31	.775015	100	0
93	.012313	1.58875	-35.9789
372	.134617	17.3697	-15.2042
434	.199398	25.7282	-11.7918
496	9.86796E-2	12.7326	-17.9016
558	.172072	22.2025	-13.072
620	5.65064E-2	7.29101	-22.7442
682	1.11796E-2	1.4425	-36.8177
713	1.36635E-2	1.76299	-35.075
775	8.57594E-2	11.0655	-19.1206
837	6.30499E-2	8.13531	-21.7925
899	7.02188E-2	9.06032	-20.8571

---

15 PWM INCREMENTS PER OUTPUT CYCLE
5 INTEGRATION STEPS PER INCREMENT
32 HZ OUTPUT FREQUENCY
PWM INCREMENTS CENTERED

COMPONENT	ABS AMPLITUDE	P.C. REL FUND	DB REL FUND
32	.775015	100	0
96	.012313	1.58875	-35.9789
384	.134617	17.3697	-15.2042
448	.199398	25.7282	-11.7918
512	9.86796E-2	12.7326	-17.9016
576	.172072	22.2025	-13.072
640	5.65064E-2	7.29101	-22.7442
704	1.11796E-2	1.4425	-36.8177
736	1.36635E-2	1.76299	-35.075
800	8.57594E-2	11.0655	-19.1206
864	6.30499E-2	8.13531	-21.7925
928	7.02188E-2	9.06032	-20.8571

---

15 PWM INCREMENTS PER OUTPUT CYCLE
5 INTEGRATION STEPS PER INCREMENT
33 HZ OUTPUT FREQUENCY
PWM INCREMENTS CENTERED

COMPONENT	ABS AMPLITUDE	P.C. REL FUND	DB REL FUND
33	.775015	100	0
99	.012313	1.58875	-35.9789
396	.134617	17.3697	-15.2042
462	.199398	25.7282	-11.7918
528	9.86796E-2	12.7326	-17.9016
594	.172072	22.2025	-13.072
660	5.65064E-2	7.29101	-22.7442
726	1.11796E-2	1.4425	-36.8177
759	1.36635E-2	1.76299	-35.075
825	8.57594E-2	11.0655	-19.1206
891	6.30499E-2	8.13531	-21.7925
957	7.02188E-2	9.06032	-20.8571

---

15 PWM INCREMENTS PER OUTPUT CYCLE
5 INTEGRATION STEPS PER INCREMENT
34 HZ OUTPUT FREQUENCY
PWM INCREMENTS CENTERED

COMPONENT	ABS AMPLITUDE	P.C. REL FUND	DB REL FUND
34	.775015	100	0
102	.012313	1.58875	-35.9789
408	.134617	17.3697	-15.2042
476	.199398	25.7282	-11.7918
544	9.86796E-2	12.7326	-17.9016
612	.172072	22.2025	-13.072
680	5.65064E-2	7.29101	-22.7442
748	1.11796E-2	1.4425	-36.8177
782	1.36635E-2	1.76299	-35.075
850	8.57594E-2	11.0655	-19.1206
918	6.30499E-2	8.13531	-21.7925
986	7.02188E-2	9.06032	-20.8571

15 PWM INCREMENTS PER OUTPUT CYCLE
5 INTEGRATION STEPS PER INCREMENT
35 HZ OUTPUT FREQUENCY
PWM INCREMENTS CENTERED

COMPONENT	ABS AMPLITUDE	P.C. REL FUND	DB REL FUND
35	.775015	100	0
105	.012313	1.58875	-35.9789
420	.134617	17.3697	-15.2042
490	.199398	25.7282	-11.7918
560	9.86796E-2	12.7326	-17.9016
630	.172072	22.2025	-13.072
700	5.65064E-2	7.29101	-22.7442
770	1.11796E-2	1.4425	-36.8177
805	1.36635E-2	1.76299	-35.075
875	8.57594E-2	11.0655	-19.1206
945	6.30499E-2	8.13531	-21.7925
1015	7.02188E-2	9.06032	-20.8571

15 PWM INCREMENTS PER OUTPUT CYCLE
5 INTEGRATION STEPS PER INCREMENT
36 HZ OUTPUT FREQUENCY
PWM INCREMENTS CENTERED

COMPONENT	ABS AMPLITUDE	P.C. REL FUND	DB REL FUND
36	.775015	100	0
108	.012313	1.58875	-35.9789
432	.134617	17.3697	-15.2042
504	.199398	25.7282	-11.7918
576	9.86796E-2	12.7326	-17.9016
648	.172072	22.2025	-13.072
720	5.65064E-2	7.29101	-22.7442
792	1.11796E-2	1.4425	-36.8177
828	1.36635E-2	1.76299	-35.075
900	8.57594E-2	11.0655	-19.1206
972	6.30499E-2	8.13531	-21.7925
1044	7.02188E-2	9.06032	-20.8571

12 PWM INCREMENTS PER OUTPUT CYCLE
5 INTEGRATION STEPS PER INCREMENT
37 HZ OUTPUT FREQUENCY
PWM INCREMENTS CENTERED

COMPONENT	ABS AMPLITUDE	P.C. REL FUND	DB REL FUND
37	.769282	100	0
111	1.87308E-2	2.43484	-32.2706
333	.123258	16.0225	-15.9054
407	.215324	27.9903	-11.0599
481	9.01056E-2	11.7129	-18.6267
555	.169775	22.0693	-13.1242
629	7.31866E-2	9.51363	-20.4331
703	9.42229E-2	12.2482	-18.2386
777	8.11422E-2	10.5478	-19.5368
851	7.43328E-2	9.66263	-20.2981
925	3.84122E-2	4.99326	-26.0323
999	2.34218E-2	3.04463	-30.3293
1073	.11725	15.2415	-16.3395

---

12 PWM INCREMENTS PER OUTPUT CYCLE
5 INTEGRATION STEPS PER INCREMENT
38 HZ OUTPUT FREQUENCY
PWM INCREMENTS CENTERED

COMPONENT	ABS AMPLITUDE	P.C. REL FUND	DB REL FUND
38	.769282	100	0
114	1.87308E-2	2.43484	-32.2706
342	.123258	16.0225	-15.9054
418	.215324	27.9903	-11.0599
494	9.01056E-2	11.7129	-18.6267
570	.169775	22.0693	-13.1242
646	7.31866E-2	9.51363	-20.4331
722	9.42229E-2	12.2482	-18.2386
798	8.11422E-2	10.5478	-19.5368
874	7.43328E-2	9.66263	-20.2981
950	3.84122E-2	4.99326	-26.0323
1026	2.34218E-2	3.04463	-30.3293
1102	.11725	15.2415	-16.3395

---

12 PWM INCREMENTS PER OUTPUT CYCLE
5 INTEGRATION STEPS PER INCREMENT
39 HZ OUTPUT FREQUENCY
PWM INCREMENTS CENTERED

COMPONENT	ABS AMPLITUDE	P.C. REL FUND	DB REL FUND
39	.769282	100	0
117	1.87308E-2	2.43484	-32.2706
351	.123258	16.0225	-15.9054
429	.215324	27.9903	-11.0599
507	9.01056E-2	11.7129	-18.6267
585	.169775	22.0693	-13.1242
663	7.31866E-2	9.51363	-20.4331
741	9.42229E-2	12.2482	-18.2386
819	8.11422E-2	10.5478	-19.5368
897	7.43328E-2	9.66263	-20.2981
975	3.84122E-2	4.99326	-26.0323
1053	2.34218E-2	3.04463	-30.3293
1131	.11725	15.2415	-16.3395

---

12 PWM INCREMENTS PER OUTPUT CYCLE
5 INTEGRATION STEPS PER INCREMENT
40 HZ OUTPUT FREQUENCY
PWM INCREMENTS CENTERED

COMPONENT	ABS AMPLITUDE	P.C. REL FUND	DB REL FUND
40	.769282	100	0
120	1.87308E-2	2.43484	-32.2706
360	.123258	16.0225	-15.9054
440	.215324	27.9903	-11.0599
520	9.01056E-2	11.7129	-18.6267
600	.169775	22.0693	-13.1242
680	7.31866E-2	9.51363	-20.4331
760	9.42229E-2	12.2482	-18.2386
840	8.11422E-2	10.5478	-19.5368
920	7.43328E-2	9.66263	-20.2981
1000	3.84122E-2	4.99326	-26.0323
1080	2.34218E-2	3.04463	-30.3293
1160	.11725	15.2415	-16.3395

---

12 PWM INCREMENTS PER OUTPUT CYCLE
5 INTEGRATION STEPS PER INCREMENT
41 HZ OUTPUT FREQUENCY
PWM INCREMENTS CENTERED

COMPONENT	ABS AMPLITUDE	P.C. REL FUND	DB REL FUND
41	.769282	100	0
123	1.87308E-2	2.43484	-32.2706
369	.123258	16.0225	-15.9054
451	.215324	27.9903	-11.0599
533	9.01056E-2	11.7129	-18.6267
615	.169775	22.0693	-13.1242
697	7.31866E-2	9.51363	-20.4331
779	9.42229E-2	12.2482	-18.2386
861	8.11422E-2	10.5478	-19.5368
943	7.43328E-2	9.66263	-20.2981
1025	3.84122E-2	4.99326	-26.0323
1107	2.34218E-2	3.04463	-30.3293
1189	.11725	15.2415	-16.3395

---

12 PWM INCREMENTS PER OUTPUT CYCLE
5 INTEGRATION STEPS PER INCREMENT
42 HZ OUTPUT FREQUENCY
PWM INCREMENTS CENTERED

COMPONENT	ABS AMPLITUDE	P.C. REL FUND	DB REL FUND
42	.769282	100	0
126	1.87308E-2	2.43484	-32.2706
378	.123258	16.0225	-15.9054
462	.215324	27.9903	-11.0599
546	9.01056E-2	11.7129	-18.6267
630	.169775	22.0693	-13.1242
714	7.31866E-2	9.51363	-20.4331
798	9.42229E-2	12.2482	-18.2386
882	8.11422E-2	10.5478	-19.5368
966	7.43328E-2	9.66263	-20.2981
1050	3.84122E-2	4.99326	-26.0323
1134	2.34218E-2	3.04463	-30.3293
1218	.11725	15.2415	-16.3395

HARMONIC ANALYSIS OF SYNTHESIZED SINE WAVE USING
CONSTANT AMPLITUDE PWM WAVEFORM.

12 PWM INCREMENTS PER OUTPUT CYCLE
5 INTEGRATION STEPS PER INCREMENT
43 HZ OUTPUT FREQUENCY
PWM INCREMENTS CENTERED

COMPONENT	ABS AMPLITUDE	P.C. REL FUND	DB REL FUND
43	.769282	100	0
129	1.87308E-2	2.43484	-32.2706
387	.123258	16.0225	-15.9054
473	.215324	27.9903	-11.0599
559	9.01056E-2	11.7129	-18.6267
645	.169775	22.0693	-13.1242
731	7.31866E-2	9.51363	-20.4331
817	9.42229E-2	12.2482	-18.2386

12 PWM INCREMENTS PER OUTPUT CYCLE
5 INTEGRATION STEPS PER INCREMENT
44 HZ OUTPUT FREQUENCY
PWM INCREMENTS CENTERED

COMPONENT	ABS AMPLITUDE	P.C. REL FUND	DB REL FUND
44	.769282	100	0
132	1.87308E-2	2.43484	-32.2706
396	.123258	16.0225	-15.9054
484	.215324	27.9903	-11.0599
572	9.01056E-2	11.7129	-18.6267
660	.169775	22.0693	-13.1242
748	7.31866E-2	9.51363	-20.4331
836	9.42229E-2	12.2482	-18.2386

---

12 PWM INCREMENTS PER OUTPUT CYCLE
5 INTEGRATION STEPS PER INCREMENT
45 HZ OUTPUT FREQUENCY
PWM INCREMENTS CENTERED

COMPONENT	ABS AMPLITUDE	P.C. REL FUND	DB REL FUND
45	.769282	100	0
135	1.87308E-2	2.43484	-32.2706
405	.123258	16.0225	-15.9054
495	.215324	27.9903	-11.0599
585	9.01056E-2	11.7129	-18.6267
675	.169775	22.0693	-13.1242
765	7.31866E-2	9.51363	-20.4331
855	9.42229E-2	12.2482	-18.2386

---

9 PWM INCREMENTS PER OUTPUT CYCLE
5 INTEGRATION STEPS PER INCREMENT
46 HZ OUTPUT FREQUENCY
PWM INCREMENTS CENTERED

COMPONENT	ABS AMPLITUDE	P.C. REL FUND	DB REL FUND
46	.757151	100	0
138	.031447	4.15333	-27.6321
276	.102612	13.5524	-17.3597
368	.242866	32.0762	-9.87634
460	7.79043E-2	10.2891	-19.7524
552	.162475	21.4587	-13.3679
598	6.46243E-2	8.53519	-21.3757
644	7.46073E-2	9.85369	-20.128
690	.102706	13.5648	-17.3517
736	2.33374E-2	3.08226	-30.2226
782	8.21435E-2	10.849	-19.2922
874	3.72516E-2	4.91996	-26.1608
920	4.69544E-2	6.20146	-24.1501

---

9 PWM INCREMENTS PER OUTPUT CYCLE
5 INTEGRATION STEPS PER INCREMENT
47 HZ OUTPUT FREQUENCY
PWM INCREMENTS CENTERED

COMPONENT	ABS AMPLITUDE	P.C. REL FUND	DB REL FUND
47	.757151	100	0
141	.031447	4.15333	-27.6321
282	.102612	13.5524	-17.3597
376	.242866	32.0762	-9.87634
470	7.79043E-2	10.2891	-19.7524
564	.162475	21.4587	-13.3679
611	6.46243E-2	8.53519	-21.3757
658	7.46073E-2	9.85369	-20.128
705	.102706	13.5648	-17.3517
752	2.33374E-2	3.08226	-30.2226
799	8.21435E-2	10.849	-19.2922
893	3.72516E-2	4.91996	-26.1608
940	4.69544E-2	6.20146	-24.1501

---

9 PWM INCREMENTS PER OUTPUT CYCLE
5 INTEGRATION STEPS PER INCREMENT
48 HZ OUTPUT FREQUENCY
PWM INCREMENTS CENTERED

COMPONENT	ABS AMPLITUDE	P.C. REL FUND	DB REL FUND
48	.757151	100	0
144	.031447	4.15333	-27.6321
288	.102612	13.5524	-17.3597
384	.242866	32.0762	-9.87634
480	7.79043E-2	10.2891	-19.7524
576	.162475	21.4587	-13.3679
624	6.46243E-2	8.53519	-21.3757
672	7.46073E-2	9.85369	-20.128
720	.102706	13.5648	-17.3517
768	2.33374E-2	3.08226	-30.2226
816	8.21435E-2	10.849	-19.2922
912	3.72516E-2	4.91996	-26.1608
960	4.69544E-2	6.20146	-24.1501

---

9 PWM INCREMENTS PER OUTPUT CYCLE
5 INTEGRATION STEPS PER INCREMENT
49 HZ OUTPUT FREQUENCY
PWM INCREMENTS CENTERED

COMPONENT	ABS AMPLITUDE	P.C. REL FUND	DB REL FUND
49	.757151	100	0
147	.031447	4.15333	-27.6321
294	.102612	13.5524	-17.3597
392	.242866	32.0762	-9.87634
490	7.79043E-2	10.2891	-19.7524
588	.162475	21.4587	-13.3679
637	6.46243E-2	8.53519	-21.3757
686	7.46073E-2	9.85369	-20.128
735	.102706	13.5648	-17.3517
784	2.33374E-2	3.08226	-30.2226
833	8.21435E-2	10.849	-19.2922
931	3.72516E-2	4.91996	-26.1608
980	4.69544E-2	6.20146	-24.1501

```
9  *INTERRUPTED*
140 H=INT(850/S)
45 FOR S=50 TO 60
RUN

 78/12/22. 09.48.48.
PROGRAM   HARM
```

HARMONIC ANALYSIS OF SYNTHESIZED SINE WAVE USING
CONSTANT AMPLITUDE PWM WAVEFORM.

9 PWM INCREMENTS PER OUTPUT CYCLE
5 INTEGRATION STEPS PER INCREMENT
50 HZ OUTPUT FREQUENCY
PWM INCREMENTS CENTERED

COMPONENT	ABS AMPLITUDE	P.C. REL FUND	DB REL FUND
50	.757151	100	0
150	.031447	4.15333	-27.6321
300	.102612	13.5524	-17.3597
400	.242866	32.0762	-9.87634
500	7.79043E-2	10.2891	-19.7524
600	.162475	21.4587	-13.3679
650	6.46243E-2	8.53519	-21.3757
700	7.46073E-2	9.85369	-20.128
750	.102706	13.5648	-17.3517
800	2.33374E-2	3.08226	-30.2226
850	8.21435E-2	10.849	-19.2922

---

9 PWM INCREMENTS PER OUTPUT CYCLE
5 INTEGRATION STEPS PER INCREMENT
51 HZ OUTPUT FREQUENCY
PWM INCREMENTS CENTERED

COMPONENT	ABS AMPLITUDE	P.C. REL FUND	DB REL FUND
51	.757151	100	0
153	.031447	4.15333	-27.6321
306	.102612	13.5524	-17.3597
408	.242866	32.0762	-9.87634
510	7.79043E-2	10.2891	-19.7524
612	.162475	21.4587	-13.3679
663	6.46243E-2	8.53519	-21.3757
714	7.46073E-2	9.85369	-20.128
765	.102706	13.5648	-17.3517
816	2.33374E-2	3.08226	-30.2226

---

9 PWM INCREMENTS PER OUTPUT CYCLE
5 INTEGRATION STEPS PER INCREMENT
52 HZ OUTPUT FREQUENCY
PWM INCREMENTS CENTERED

COMPONENT	ABS AMPLITUDE	P.C. REL FUND	DB REL FUND
52	.757151	100	0
156	.031447	4.15333	-27.6321
312	.102612	13.5524	-17.3597
416	.242866	32.0762	-9.87634
520	7.79043E-2	10.2891	-19.7524
624	.162475	21.4587	-13.3679
676	6.46243E-2	8.53519	-21.3757
728	7.46073E-2	9.85369	-20.128
780	.102706	13.5648	-17.3517
832	2.33374E-2	3.08226	-30.2226

---

9 PWM INCREMENTS PER OUTPUT CYCLE
5 INTEGRATION STEPS PER INCREMENT
53 HZ OUTPUT FREQUENCY
PWM INCREMENTS CENTERED

COMPONENT	ABS AMPLITUDE	P.C. REL FUND	DB REL FUND
53	.757151	100	0
159	.031447	4.15333	-27.6321
318	.102612	13.5524	-17.3597
424	.242866	32.0762	-9.87634
530	7.79043E-2	10.2891	-19.7524
636	.162475	21.4587	-13.3679
689	6.46243E-2	8.53519	-21.3757
742	7.46073E-2	9.85369	-20.128
795	.102706	13.5648	-17.3517
848	2.33374E-2	3.08226	-30.2226

---

9 PWM INCREMENTS PER OUTPUT CYCLE
5 INTEGRATION STEPS PER INCREMENT
54 HZ OUTPUT FREQUENCY
PWM INCREMENTS CENTERED

COMPONENT	ABS AMPLITUDE	P.C. REL FUND	DB REL FUND
54	.757151	100	0
162	.031447	4.15333	-27.6321
324	.102612	13.5524	-17.3597
432	.242866	32.0762	-9.87634
540	7.79043E-2	10.2891	-19.7524
648	.162475	21.4587	-13.3679
702	6.46243E-2	8.53519	-21.3757
756	7.46073E-2	9.85369	-20.128
810	.102706	13.5648	-17.3517

---

9 PWM INCREMENTS PER OUTPUT CYCLE
5 INTEGRATION STEPS PER INCREMENT
55 HZ OUTPUT FREQUENCY
PWM INCREMENTS CENTERED

COMPONENT	ABS AMPLITUDE	P.C. REL FUND	DB REL FUND
55	.757151	100	0
165	.031447	4.15333	-27.6321
330	.102612	13.5524	-17.3597
440	.242866	32.0762	-9.87634
550	7.79043E-2	10.2891	-19.7524
660	.162475	21.4587	-13.3679
715	6.46243E-2	8.53519	-21.3757
770	7.46073E-2	9.85369	-20.128
825	.102706	13.5648	-17.3517

---

9 PWM INCREMENTS PER OUTPUT CYCLE
5 INTEGRATION STEPS PER INCREMENT
56 HZ OUTPUT FREQUENCY
PWM INCREMENTS CENTERED

COMPONENT	ABS AMPLITUDE	P.C. REL FUND	DB REL FUND
56	.757151	100	0
168	.031447	4.15333	-27.6321
336	.102612	13.5524	-17.3597
448	.242866	32.0762	-9.87634
560	7.79043E-2	10.2891	-19.7524
672	.162475	21.4587	-13.3679
728	6.46243E-2	8.53519	-21.3757
784	7.46073E-2	9.85369	-20.128
840	.102706	13.5648	-17.3517

---

9 PWM INCREMENTS PER OUTPUT CYCLE
5 INTEGRATION STEPS PER INCREMENT
57 HZ OUTPUT FREQUENCY
PWM INCREMENTS CENTERED

COMPONENT	ABS AMPLITUDE	P.C. REL FUND	DB REL FUND
57	.757151	100	0
171	.031447	4.15333	-27.6321
342	.102612	13.5524	-17.3597
456	.242866	32.0762	-9.87634
570	7.79043E-2	10.2891	-19.7524
684	.162475	21.4587	-13.3679
741	6.46243E-2	8.53519	-21.3757
798	7.46073E-2	9.85369	-20.128

---

9 PWM INCREMENTS PER OUTPUT CYCLE
5 INTEGRATION STEPS PER INCREMENT
58 HZ OUTPUT FREQUENCY
PWM INCREMENTS CENTERED

COMPONENT	ABS AMPLITUDE	P.C. REL FUND	DB REL FUND
58	.757151	100	0
174	.031447	4.15333	-27.6321
348	.102612	13.5524	-17.3597
464	.242866	32.0762	-9.87634
580	7.79043E-2	10.2891	-19.7524
696	.162475	21.4587	-13.3679
754	6.46243E-2	8.53519	-21.3757
812	7.46073E-2	9.85369	-20.128

9 PWM INCREMENTS PER OUTPUT CYCLE
5 INTEGRATION STEPS PER INCREMENT
59 HZ OUTPUT FREQUENCY
PWM INCREMENTS CENTERED

COMPONENT	ABS AMPLITUDE	P.C. REL FUND	DB REL FUND
59	.757151	100	0
177	.031447	4.15333	-27.6321
354	.102612	13.5524	-17.3597
472	.242866	32.0762	-9.87634
590	7.79043E-2	10.2891	-19.7524
708	.162475	21.4587	-13.3679
767	6.46243E-2	8.53519	-21.3757
826	7.46073E-2	9.85369	-20.128

9 PWM INCREMENTS PER OUTPUT CYCLE
5 INTEGRATION STEPS PER INCREMENT
60 HZ OUTPUT FREQUENCY
PWM INCREMENTS CENTERED

COMPONENT	ABS AMPLITUDE	P.C. REL FUND	DB REL FUND
60	.757151	100	0
180	.031447	4.15333	-27.6321
360	.102612	13.5524	-17.3597
480	.242866	32.0762	-9.87634
600	7.79043E-2	10.2891	-19.7524
720	.162475	21.4587	-13.3679
780	6.46243E-2	8.53519	-21.3757
840	7.46073E-2	9.85369	-20.128

APPENDIX D

Average Values of PWM Steps - Computer Ouput

```
AVERAGE VALUE OF SINE WAVE OVER SPECIFIED INTERVAL .
FORMULATION:
            COS(X1)-COS(X2)
   G(X) = -----------------
              (X1-X2)
```

9.000 INTERVALS

INTERVAL (DEGREES) FROM	TO	AVERAGE VALUE
0.000	40.000	0.335
40.000	80.000	0.849
80.000	120.000	0.965
120.000	160.000	0.630
160.000	200.000	0.000
200.000	240.000	-0.630
240.000	280.000	-0.965
280.000	320.000	-0.849
320.000	360.000	-0.335

15.[illegible] INTERVALS

INTERVAL (DEGREES) FROM	TO	AVERAGE VALUE
0.000	24.000	0.206
24.000	48.000	0.583
48.000	72.000	0.860
72.000	96.000	0.987
96.000	120.000	0.944
120.000	144.000	0.738
144.000	168.000	0.40[illegible]
168.000	192.000	0.000
192.000	216.000	[illegible]0.404
216.000	240.000	-0.738
240.000	264.000	-0.944
264.000	288.000	-0.987
288.000	312.000	-0.860
312.000	336.000	-0.583
336.000	360.000	-0.206

INTERVAL (DEGREES) AVERAGE VALUE — 21.000 INTERVALS

FROM	TO	AVERAGE VALUE
0.000	17.143	0.148
17.143	34.286	0.432
34.286	51.429	0.678
51.429	68.571	0.863
68.571	85.714	0.971
85.714	102.857	0.993
102.857	120.000	0.927
120.000	137.143	0.779
137.143	154.286	0.561
154.286	171.429	0.294
171.429	188.571	0.000
188.571	205.714	-0.294
205.714	222.857	-0.561
222.857	240.000	-0.779
240.000	257.143	-0.927
257.143	274.286	-0.993
274.286	291.429	-0.971
291.429	308.571	-0.863
308.571	325.714	-0.678
325.714	342.857	-0.432
342.857	360.000	-0.148

INTERVAL (DEGREES) AVERAGE VALUE — 27.000 INTERVALS

FROM	TO	AVERAGE VALUE
0.000	13.333	0.116
13.333	26.667	0.341
26.667	40.000	0.548
40.000	53.333	0.726
53.333	66.667	0.864
66.667	80.000	0.956
80.000	93.333	0.996
93.333	106.667	0.983
106.667	120.000	0.916
120.000	133.333	0.800
133.333	146.667	0.641
146.667	160.000	0.448
160.000	173.333	0.230
173.333	186.667	0.000
186.667	200.000	-0.230
200.000	213.333	-0.448
213.333	226.667	-0.641
226.667	240.000	-0.800
240.000	253.333	-0.916
253.333	266.667	-0.983
266.667	280.000	-0.996
280.000	293.333	-0.956
293.333	306.667	-0.864
306.667	320.000	-0.726
320.000	333.333	-0.548
333.333	346.667	-0.341
346.667	360.000	-0.116

33.000 INTERVALS

INTERVAL (DEGREES) FROM	TO	AVERAGE VALUE
0.000	10.909	0.095
10.909	21.818	0.281
21.818	32.727	0.458
32.727	43.636	0.617
43.636	54.545	0.755
54.545	65.455	0.865
65.455	76.364	0.944
76.364	87.273	0.988
87.273	98.182	0.997
98.182	109.091	0.970
109.091	120.000	0.908
120.000	130.909	0.813
130.909	141.818	0.689
141.818	152.727	0.540
152.727	163.636	0.371
163.636	174.545	0.189
174.545	185.455	0.000
185.455	196.364	-0.189
196.364	207.273	-0.371
207.273	218.182	-0.540
218.182	229.091	-0.689
229.091	240.000	-0.813
240.000	250.909	-0.908
250.909	261.818	-0.970
261.818	272.727	-0.997
272.727	283.636	-0.988
283.636	294.545	-0.944
294.545	305.455	-0.865
305.455	316.364	-0.755
316.364	327.273	-0.617
327.273	338.182	-0.458
338.182	349.091	-0.281
349.091	360.000	-0.095

[illegible] INTERVALS

INTERVAL (DEGREES) FROM	TO	AVERAGE VALUE
0.000	9.231	0.080
9.231	18.462	0.239
18.462	27.692	0.392
27.692	36.923	0.534
36.923	46.154	0.662
46.154	55.385	0.774
55.385	64.615	0.865
64.615	73.846	0.934
73.846	83.077	0.979
83.077	92.308	0.998
92.308	101.538	0.992
101.538	110.769	0.959
110.769	120.000	0.902
120.000	129.231	0.822
129.231	138.462	0.720
138.462	147.692	0.600
147.692	156.923	0.464
156.923	166.154	0.316
166.154	175.385	0.160
175.385	184.615	0.000
184.615	193.846	-0.160
193.846	203.077	-0.316
203.077	212.308	-0.464
212.308	221.538	-0.600
221.538	230.769	-0.720
230.769	240.000	-0.822
240.000	249.231	-0.902
249.231	258.462	-0.959
258.462	267.692	-0.992
267.692	276.923	-0.998
276.923	286.154	-0.979
286.154	295.385	-0.934
295.385	304.615	-0.865
304.615	313.846	-0.774
313.846	323.077	-0.662
323.077	332.308	-0.534
332.308	341.538	-0.392
341.538	350.769	-0.239

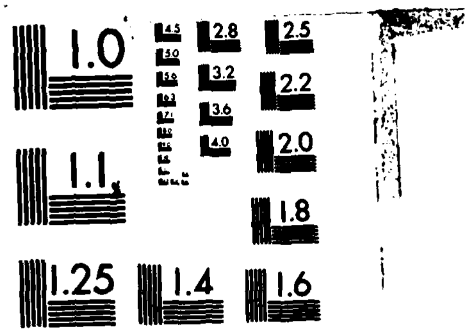
INTERVAL (DEGREES) FROM	TO	AVERAGE VALUE
0.000	8.000	0.070
8.000	16.000	0.208
16.000	24.000	0.342
24.000	32.000	0.469
32.000	40.000	0.587
40.000	48.000	0.694
48.000	56.000	0.787
56.000	64.000	0.865
64.000	72.000	0.926
72.000	80.000	0.970
80.000	88.000	0.994
88.000	96.000	0.999
96.000	104.000	0.984
104.000	112.000	0.950
112.000	120.000	0.898
120.000	128.000	0.828
128.000	136.000	0.743
136.000	144.000	0.642
144.000	152.000	0.529
152.000	160.000	0.406
160.000	168.000	0.275
168.000	176.000	0.139
176.000	184.000	-0.000
184.000	192.000	-0.139
192.000	200.000	-0.275
200.000	208.000	-0.406
208.000	216.000	-0.529
216.000	224.000	-0.642
224.000	232.000	-0.743
232.000	240.000	-0.828
240.000	248.000	-0.898
248.000	256.000	-0.950
256.000	264.000	-0.984
264.000	272.000	-0.999
272.000	280.000	-0.994
280.000	288.000	-0.970
288.000	296.000	-0.926
296.000	304.000	-0.865
304.000	312.000	-0.787
312.000	320.000	-0.694
320.000	328.000	-0.587
328.000	336.000	-0.469
336.000	344.000	-0.342
344.000	352.000	-0.208
352.000	360.000	-0.070

[illegible] INTERVALS

AD-A140 299 ADVANCED MOTOR-CONTROLLER DEVELOPMENT(U) WESTINGHOUSE OCEANIC DIV ANNAPOLIS MD L E LESSTER ET AL. 22 JUN 83 DTNSRDC-PASD-CR-1-83 N00167-82-C-0122

2/4

UNCLASSIFIED F/G 9/3 NL



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

INTERVAL (DEGREES) FROM	TO	AVERAGE VALUE
0.000	7.059	0.062
7.059	14.118	0.184
14.118	21.176	0.303
21.176	28.235	0.418
28.235	35.294	0.526
35.294	42.353	0.627
42.353	49.412	0.717
49.412	56.471	0.798
56.471	63.529	0.865
63.529	70.588	0.920
70.588	77.647	0.961
77.647	84.706	0.988
84.706	91.765	0.999
91.765	98.824	0.995
98.824	105.882	0.976
105.882	112.941	0.943
112.941	120.000	0.895
120.000	127.059	0.833
127.059	134.118	0.759
134.118	141.176	0.673
141.176	148.235	0.577
148.235	155.294	0.473
155.294	162.353	0.361
162.353	169.412	0.244
169.412	176.471	0.123
176.471	183.529	-0.000
183.529	190.588	-0.123
190.588	197.647	-0.244
197.647	204.706	-0.361
204.706	211.765	-0.473
211.765	218.824	-0.577
218.824	225.882	-0.673
225.882	232.941	-0.759
232.941	240.000	-0.833
240.000	247.059	-0.895
247.059	254.118	-0.943
254.118	261.176	-0.976
261.176	268.235	-0.995
268.235	275.294	-0.999
275.294	282.353	-0.988
282.353	289.412	-0.961
289.412	296.471	-0.920
296.471	303.529	-0.865
303.529	310.588	-0.798
310.588	317.647	-0.717
317.647	324.706	-0.627
324.706	331.765	-0.526
331.765	338.824	-0.418
338.824	345.882	-0.303
345.882	352.941	-0.184

51.000 INTERVALS

[illegible] INTERVALS

INTERVAL (DEGREES) FROM	TO	AVERAGE VALUE
0.000	6.316	0.055
6.316	12.632	0.165
12.632	18.947	0.272
18.947	25.263	0.376
25.263	31.579	0.476
31.579	37.895	0.570
37.895	44.211	0.656
44.211	50.526	0.735
50.526	56.842	0.805
56.842	63.158	0.866
63.158	69.474	0.915
69.474	75.789	0.954
75.789	82.105	0.981
82.105	88.421	0.996
88.421	94.737	0.999
94.737	101.053	0.990
101.053	107.368	0.969
107.368	113.684	0.936
113.684	120.000	0.892
120.000	126.316	0.837
126.316	132.632	0.772
132.632	138.947	0.697
138.947	145.263	0.614
145.263	151.579	0.523
151.579	157.895	0.427
157.895	164.211	0.325
164.211	170.526	0.219
170.526	176.842	0.110
176.842	183.158	0.000
183.158	189.474	-0.110
189.474	195.789	-0.219
195.789	202.105	-0.325
202.105	208.421	-0.427
208.421	214.737	-0.523
214.737	221.053	-0.614
221.053	227.368	-0.697
227.368	233.684	-0.772
233.684	240.000	-0.837
240.000	246.316	-0.892
246.316	252.632	-0.936
252.632	258.947	-0.969
258.947	265.263	-0.990
265.263	271.579	-0.999
271.579	277.895	-0.996
277.895	284.211	-0.981
284.211	290.526	-0.954
290.526	296.842	-0.915
296.842	303.158	-0.866
303.158	309.474	-0.805
309.474	315.789	-0.735
315.789	322.105	-0.656
322.105	328.421	-0.570
328.421	334.737	-0.476
334.737	341.053	-0.376
341.053	347.368	-0.272
347.368	353.684	-0.165

63.000 INTERVALS

INTERVAL (DEGREES) FROM	TO	AVERAGE VALUE
0.000	5.714	0.050
5.714	11.429	0.149
11.429	17.143	0.247
17.143	22.857	0.342
22.857	28.571	0.434
28.571	34.286	0.521
34.286	40.000	0.604
40.000	45.714	0.680
45.714	51.429	0.749
51.429	57.143	0.812
57.143	62.857	0.866
62.857	68.571	0.911
68.571	74.286	0.948
74.286	80.000	0.975
80.000	85.714	0.992
85.714	91.429	0.999
91.429	97.143	0.997
97.143	102.857	0.984
102.857	108.571	0.962
108.571	114.286	0.930
114.286	120.000	0.890
120.000	125.714	0.840
125.714	131.429	0.782
131.429	137.143	0.716
137.143	142.857	0.643
142.857	148.571	0.563
148.571	154.286	0.478
154.286	160.000	0.388
160.000	165.714	0.295
165.714	171.429	0.198
171.429	177.143	0.100
177.143	182.857	0.000
182.857	188.571	-0.100
188.571	194.286	-0.198
194.286	200.000	-0.295
200.000	205.714	-0.388
205.714	211.429	-0.478
211.429	217.143	-0.563
217.143	222.857	-0.643
222.857	228.571	-0.716
228.571	234.286	-0.782
234.286	240.000	-0.840
240.000	245.714	-0.890
245.714	251.429	-0.930
251.429	257.143	-0.962
257.143	262.857	-0.984
262.857	268.571	-0.997
268.571	274.286	-0.999
274.286	280.000	-0.992
280.000	285.714	-0.975
285.714	291.429	-0.948
291.429	297.143	-0.911
297.143	302.857	-0.866
302.857	308.571	-0.812
308.571	314.286	-0.749
314.286	320.000	-0.680
320.000	325.714	-0.604
325.714	331.429	-0.521
331.429	337.143	-0.434
337.143	342.857	-0.342
342.857	348.571	-0.247
348.571	354.286	-0.149

D8

INTERVAL (DEGREES) FROM	TO	AVERAGE VALUE
0.000	5.217	0.045
5.217	10.435	0.136
10.435	15.652	0.226
15.652	20.870	0.313
20.870	26.087	0.398
26.087	31.304	0.480
31.304	36.522	0.558
36.522	41.739	0.631
41.739	46.957	0.699
46.957	52.174	0.761
52.174	57.391	0.817
57.391	62.609	0.866
62.609	67.826	0.908
67.826	73.043	0.942
73.043	78.261	0.968
78.261	83.478	0.987
83.478	88.696	0.997
88.696	93.913	0.999
93.913	99.130	0.993
99.130	104.348	0.979
104.348	109.565	0.956
109.565	114.783	0.926
114.783	120.000	0.888
120.000	125.217	0.842
125.217	130.435	0.790
130.435	135.652	0.731
135.652	140.870	0.666
140.870	146.087	0.595
146.087	151.304	0.519
151.304	156.522	0.440
156.522	161.739	0.356
161.739	166.957	0.270
166.957	172.174	0.181
172.174	177.391	0.091
177.391	182.609	-0.000
182.609	187.826	-0.091
187.826	193.043	-0.181
193.043	198.261	-0.270
198.261	203.478	-0.356
203.478	208.696	-0.440
208.696	213.913	-0.519
213.913	219.130	-0.595
219.130	224.348	-0.666
224.348	229.565	-0.731
229.565	234.783	-0.790
234.783	240.000	-0.842
240.000	245.217	-0.888
245.217	250.435	-0.926
250.435	255.652	-0.956
255.652	260.870	-0.979
260.870	266.087	-0.993
266.087	271.304	-0.999
271.304	276.522	-0.997
276.522	281.739	-0.987
281.739	286.957	-0.968
286.957	292.174	-0.942
292.174	297.391	-0.908

69.000 INTERVALS

297.391	302.609	-0.866
302.609	307.826	-0.817
307.826	313.043	-0.761
313.043	318.261	-0.699
318.261	323.478	-0.631
323.478	328.696	-0.558
328.696	333.913	-0.480
333.913	339.130	-0.398
339.130	344.348	-0.313
344.348	349.565	-0.226
349.565	354.783	-0.136

75.000 INTERVALS

INTERVAL (DEGREES) FROM	TO	AVERAGE VALUE
0.000	4.800	0.042
4.800	9.600	0.125
9.600	14.400	0.208
14.400	19.200	0.289
19.200	24.000	0.368
24.000	28.800	0.445
28.800	33.600	0.518
33.600	38.400	0.588
38.400	43.200	0.653
43.200	48.000	0.714
48.000	52.800	0.770
52.800	57.600	0.821
57.600	62.400	0.866
62.400	67.200	0.905
67.200	72.000	0.937
72.000	76.800	0.963
76.800	81.600	0.982
81.600	86.400	0.994
86.400	91.200	0.999
91.200	96.000	0.998
96.000	100.800	0.989
100.800	105.600	0.973
105.600	110.400	0.951
110.400	115.200	0.922
115.200	120.000	0.886
120.000	124.800	0.844
124.800	129.600	0.796
129.600	134.400	0.743
134.400	139.200	0.684
139.200	144.000	0.621
144.000	148.800	0.553
148.800	153.600	0.482
153.600	158.400	0.407
158.400	163.200	0.329
163.200	168.000	0.249
168.000	172.800	0.167
172.800	177.600	0.084
177.600	182.400	-0.000
182.400	187.200	-0.084
187.200	192.000	-0.167
192.000	196.800	-0.249
196.800	201.600	-0.329

201.600	206.400	-0.407
206.400	211.200	-0.482
211.200	216.000	-0.553
216.000	220.800	-0.621
220.800	225.600	-0.684
225.600	230.400	-0.743
230.400	235.200	-0.796
235.200	240.000	0.844
240.000	244.800	-0.886
244.800	249.600	-0.922
249.600	254.400	-0.951
254.400	259.200	-0.973
259.200	264.000	-0.989
264.000	268.800	-0.998
268.800	273.600	-0.999
273.600	278.400	-0.994
278.400	283.200	-0.982
283.200	288.000	-0.963
288.000	292.800	-0.937
292.800	297.600	-0.905
297.600	302.400	-0.866
302.400	307.200	-0.821
307.200	312.000	-0.770
312.000	316.800	-0.714
316.800	321.600	-0.653
321.600	326.400	-0.588
326.400	331.200	-0.518
331.200	336.000	-0.445
336.000	340.800	-0.368
340.800	345.600	-0.289
345.600	350.400	-0.208
350.400	355.200	-0.125

81.000 INTERVALS

INTERVAL (DEGREES) FROM	TO	AVERAGE VALUE
0.000	4.444	0.039
4.444	8.889	0.116
8.889	13.333	0.193
13.333	17.778	0.268
17.778	22.222	0.342
22.222	26.667	0.414
26.667	31.111	0.483
31.111	35.556	0.549
35.556	40.000	0.612
40.000	44.444	0.672
44.444	48.889	0.727
48.889	53.333	0.778
53.333	57.778	0.824
57.778	62.222	0.866
62.222	66.667	0.902
66.667	71.111	0.933
71.111	75.556	0.958
75.556	80.000	0.977
80.000	84.444	0.991
84.444	88.889	0.998
88.889	93.333	1.000

93.333	97.778	0.995
97.778	102.222	0.985
102.222	106.667	0.968
106.667	111.111	0.946
111.111	115.556	0.918
115.556	120.000	0.885
120.000	124.444	0.846
124.444	128.889	0.802
128.889	133.333	0.753
133.333	137.778	0.700
137.778	142.222	0.643
142.222	146.667	0.581
146.667	151.111	0.517
151.111	155.556	0.449
155.556	160.000	0.378
160.000	164.444	0.305
164.444	168.889	0.231
168.889	173.333	0.154
173.333	177.778	0.077
177.778	182.222	-0.000
182.222	186.667	-0.077
186.667	191.111	-0.154
191.111	195.556	-0.231
195.556	200.000	-0.305
200.000	204.444	-0.378
204.444	208.889	-0.449
208.889	213.333	-0.517
213.333	217.778	-0.581
217.778	222.222	-0.643
222.222	226.667	-0.700
226.667	231.111	-0.753
231.111	235.556	-0.802
235.556	240.000	-0.846
240.000	244.444	-0.885
244.444	248.889	-0.918
248.889	253.333	-0.946
253.333	257.778	-0.968
257.778	262.222	-0.985
262.222	266.667	-0.995
266.667	271.111	-1.000
271.111	275.556	-0.998
275.556	280.000	-0.991
280.000	284.444	-0.977
284.444	288.889	-0.958
288.889	293.333	-0.933
293.333	297.778	-0.902
297.778	302.222	-0.866
302.222	306.667	-0.824
306.667	311.111	-0.778
311.111	315.556	-0.727
315.556	320.000	-0.672
320.000	324.444	-0.612
324.444	328.889	-0.549
328.889	333.333	-0.483
333.333	337.778	-0.414
337.778	342.222	-0.342
342.222	346.667	-0.268
346.667	351.111	-0.193
351.111	355.556	-0.116

87.000 INTERVALS

INTERVAL (DEGREES) FROM	TO	AVERAGE VALUE
0.000	4.138	0.036
4.138	8.276	0.108
8.276	12.414	0.180
12.414	16.552	0.250
16.552	20.690	0.319
20.690	24.828	0.387
24.828	28.966	0.452
28.966	33.103	0.515
33.103	37.241	0.576
37.241	41.379	0.633
41.379	45.517	0.688
45.517	49.655	0.738
49.655	53.793	0.785
53.793	57.931	0.828
57.931	62.069	0.866
62.069	66.207	0.900
66.207	70.345	0.929
70.345	74.483	0.953
74.483	78.621	0.972
78.621	82.759	0.987
82.759	86.897	0.996
86.897	91.034	1.000
91.034	95.172	0.998
95.172	99.310	0.992
99.310	103.448	0.980
103.448	107.586	0.963
107.586	111.724	0.942
111.724	115.862	0.915
115.862	120.000	0.883
120.000	124.138	0.847
124.138	128.276	0.807
128.276	132.414	0.762
132.414	136.552	0.713
136.552	140.690	0.661
140.690	144.828	0.605
144.828	148.966	0.546
148.966	153.103	0.484
153.103	157.241	0.420
157.241	161.379	0.353
161.379	165.517	0.285
165.517	169.655	0.215
169.655	173.793	0.144
173.793	177.931	0.072
177.931	182.069	0.000
182.069	186.207	-0.072
186.207	190.345	-0.144
190.345	194.483	-0.215
194.483	198.621	-0.285
198.621	202.759	-0.353
202.759	206.897	-0.420
206.897	211.034	-0.484
211.034	215.172	-0.546
215.172	219.310	-0.605
219.310	223.448	-0.661
223.448	227.586	-0.713
227.586	231.724	-0.762
231.724	235.862	-0.807
235.862	240.000	-0.847

D13

[illegible]	[illegible]	[illegible]
244.138	248.276	-0.915
248.276	252.414	-0.942
252.414	256.552	-0.963
256.552	260.690	-0.980
260.690	264.828	-0.992
264.828	268.966	-0.998
268.966	273.103	-1.000
273.103	277.241	-0.996
277.241	281.379	-0.987
281.379	285.517	-0.972
285.517	289.655	-0.953
289.655	293.793	-0.929
293.793	297.931	-0.900
297.931	302.069	-0.866
302.069	306.207	-0.828
306.207	310.345	-0.785
310.345	314.483	-0.738
314.483	318.621	-0.688
318.621	322.759	-0.633
322.759	326.897	-0.576
326.897	331.034	-0.515
331.034	335.172	-0.452
335.172	339.310	-0.387
339.310	343.448	-0.319
343.448	347.586	-0.250
347.586	351.724	-0.180
351.724	355.862	-0.108
355.862	360.000	-0.036

93.000 INTERVALS

INTERVAL (DEGREES) FROM	TO	AVERAGE VALUE
0.000	3.871	0.034
3.871	7.742	0.101
7.742	11.613	0.168
11.613	15.484	0.234
15.484	19.355	0.299
19.355	23.226	0.363
23.226	27.097	0.425
27.097	30.968	0.485
30.968	34.839	0.543
34.839	38.710	0.599
38.710	42.581	0.651
42.581	46.452	0.701
46.452	50.323	0.748
50.323	54.194	0.791
54.194	58.065	0.830
58.065	61.935	0.866
61.935	65.806	0.898
65.806	69.677	0.925
69.677	73.548	0.949
73.548	77.419	0.968
77.419	81.290	0.983
81.290	85.161	0.993
85.161	89.032	0.999
89.032	92.903	1.000
92.903	96.774	0.996
96.774	100.645	0.988
100.645	104.516	0.976
104.516	108.387	0.959
108.387	112.258	0.938

112.258	116.129	0.912
116.129	120.000	0.882
120.000	123.871	0.848
123.871	127.742	0.811
127.742	131.613	0.770
131.613	135.484	0.725
135.484	139.355	0.676
139.355	143.226	0.625
143.226	147.097	0.571
147.097	150.968	0.514
150.968	154.839	0.455
154.839	158.710	0.394
158.710	162.581	0.331
162.581	166.452	0.267
166.452	170.323	0.201
170.323	174.194	0.135
174.194	178.065	0.067
178.065	181.935	-0.000
181.935	185.806	-0.067
185.806	189.677	-0.135
189.677	193.548	-0.201
193.548	197.419	-0.267
197.419	201.290	-0.331
201.290	205.161	-0.394
205.161	209.032	-0.455
209.032	212.903	-0.514
212.903	216.774	-0.571
216.774	220.645	-0.625
220.645	224.516	-0.676
224.516	228.387	-0.725
228.387	232.258	-0.770
232.258	236.129	-0.811
236.129	240.000	-0.848
240.000	243.871	-0.882
243.871	247.742	-0.912
247.742	251.613	-0.938
251.613	255.484	-0.959
255.484	259.355	-0.976
259.355	263.226	-0.988
263.226	267.097	-0.996
267.097	270.968	-1.000
270.968	274.839	-0.999
274.839	278.710	-0.993
278.710	282.581	-0.983
282.581	286.452	-0.968
286.452	290.323	-0.949
290.323	294.194	-0.925
294.194	298.065	-0.898
298.065	301.935	-0.866
301.935	305.806	-0.830
305.806	309.677	-0.791
309.677	313.548	-0.748
313.548	317.419	-0.701
317.419	321.290	-0.651
321.290	325.161	-0.599
325.161	329.032	-0.543
329.032	332.903	-0.485
332.903	336.[illegible]74	-0.425
336.774	34[illegible]645	-0.363
340.645	34[illegible].516	-0.299
344.516	348.387	-0.234
348.387	352.258	-0.168
352.258	356.129	-0.101

D15

INTERVAL (DEGREES) FROM	TO	AVERAGE VALUE
0.000	3.636	0.032
3.636	7.273	0.095
7.273	10.909	0.158
10.909	14.545	0.220
14.545	18.182	0.282
18.182	21.818	0.342
21.818	25.455	0.401
25.455	29.091	0.458
29.091	32.727	0.514
32.727	36.364	0.567
36.364	40.000	0.618
40.000	43.636	0.667
43.636	47.273	0.713
47.273	50.909	0.756
50.909	54.545	0.796
54.545	58.182	0.832
58.182	61.818	0.866
61.818	65.455	0.896
65.455	69.091	0.922
69.091	72.727	0.945
72.727	76.364	0.964
76.364	80.000	0.979
80.000	83.636	0.990
83.636	87.273	0.997
87.273	90.909	1.000
90.909	94.545	0.999
94.545	98.182	0.994
98.182	101.818	0.985
101.818	105.455	0.972
105.455	109.091	0.955
109.091	112.727	0.934
112.727	116.364	0.909
116.364	120.000	0.881
120.000	123.636	0.850
123.636	127.273	0.814
127.273	130.909	0.776
130.909	134.545	0.734
134.545	138.182	0.690
138.182	141.818	0.643
141.818	145.455	0.593
145.455	149.091	0.541
149.091	152.727	0.486
152.727	156.364	0.430
156.364	160.000	0.372
160.000	163.636	0.312
163.636	167.273	0.251
167.273	170.909	0.189
170.909	174.545	0.127
174.545	178.182	0.063
178.182	181.818	0.000
181.818	185.455	-0.063
185.455	189.091	-0.127
189.091	192.727	-0.189
192.727	196.364	-0.251
196.364	200.000	-0.312

99.000 INTERVALS

200.000	203.636	-0.372
203.636	207.273	-0.430
207.273	210.909	-0.486
210.909	214.545	-0.541
214.545	218.182	-0.593
218.182	221.818	-0.643
221.818	225.455	-0.690
225.455	229.091	-0.734
229.091	232.727	-0.776
232.727	236.364	-0.814
236.364	240.000	-0.850
240.000	243.636	-0.881
243.636	247.273	-0.909
247.273	250.909	-0.934
250.909	254.545	-0.955
254.545	258.182	-0.972
258.182	261.818	-0.985
261.818	265.455	-0.994
265.455	269.091	-0.999
269.091	272.727	-1.000
272.727	276.364	-0.997
276.364	280.000	-0.990
280.000	283.636	-0.979
283.636	287.273	-0.964
287.273	290.909	-0.945
290.909	294.545	-0.922
294.545	298.182	-0.896
298.182	301.818	-0.866
301.818	305.455	-0.832
305.455	309.091	-0.796
309.091	312.727	-0.756
312.727	316.364	-0.713
316.364	320.000	-0.667
320.000	323.636	-0.618
323.636	327.273	-0.567
327.273	330.909	-0.514
330.909	334.545	-0.458
334.545	338.182	-0.401
338.182	341.818	-0.342
341.818	345.455	-0.282
345.455	349.091	-0.220
349.091	352.727	-0.158
352.727	356.364	-0.095
356.364	360.000	-0.032

ADVANCED MOTOR CONTROLLER

PHASE II REPORT

- PULSE WIDTH MODULATION AND FEEDBACK CONTROL CIRCUITS

CONTENTS

ADVANCED MOTOR CONTROLLER

PHASE II REPORT

1. The Phase I study showed that it was possible to utilize an 8 bit microprocessor (M6800) to generate 3 phase sine wave output pulse width modulation waveforms for variable frequency control, via a power bridge, of an induction motor. Based on the theoretical work of Abbondanti (Ⓦ Research and Development Center) a preliminary look was taken at air gap flux synthesis for use as a feedback function for optimum motor control with load.

This Phase II study has taken the preliminary work a step further and has developed a design for the pulse width modulation and for the feedback utilizing separate processors for each function and connected via a parallel port.

The PWM generation utilizes volt second integral determination of each pulse width and pulse by pulse modification of pulse width is possible in response to voltage control requirements for a wide control bandwidth.

The feedback processor samples the input to the motor. Since power does not fluctuate at the supply frequency, but is "invariant", asynchronous sampling of the output is possible provided that in-phase and quadrature components of current and voltage are sampled simultaneously. The derived reactive power components are calculated and a digitally smoothed error signal is passed to the PWM processor.

The error signal is used to scale the PWM processor output voltage and also as an indication of overload. Overload causes a controlled ramp down in frequency until load is reduced or the off condition is reached.

Reversing is implemented as a ramp down to zero speed followed by a ramp up in reverse rotation to the set speed.

Controls can be analog or digital, via potentiometers and switches or a serial port. A terminal can be connected to the serial port for control and diagnostic purposes using a built in software monitor routine.

2. Theoretical Basis for Control Circuit Design Approach

The Phase I report described in detail basic modulation generation techniques and also the basic calculations leading to the Air Gap Flux synthesis approach to be employed here.

The following pages serve to consolidate these facts and summarize the basis for the control circuit design.

The control circuits assume that a three phase bridge is to be driven to produce variable frequency three phase waveforms with a sinewave fundamental component as shown in Figure 2.1. The three phase waveforms repeat every 60 degrees provided allowance is made for interchange of the dominant and complementary roles of the line to line voltages. Within a 60 degree period the waveforms are symmetrical about the centerpoint provided allowance is made for interchange of the role of the two complementary waveforms. Thus by organized use of symmetry properties the Pulse Width Modulation waveforms can be generated economically.

The number of pulses per 60^{o} interval are chosen to be odd to ease symmetry requirements. A minimum of 3 pulse periods per 60^{o} interval is defined at max frequency (60 Hz) hence approximate frequency at "carrier".

$f_c = 3 \times 6 \times 60 = 1080$ Hz for $926\ \mu s$ per pulse period maximum. It should be noted that the power switches make only one <u>transition</u> each during a pulse period, thus their effective switching frequency is one half the carrier frequency or 540 Hz approximately the actual number varies as the output frequency and number of pulses per 60^{o} interval is varied.

Figure 2.2 illustrates both the volt second integral approach to pulse width determination and also the basic pulsewidths associated with a single pulse period of the several that would be present in one 60 degree interval.

Figure 2.3 extends this to the time relationship actually used in the microprocessor software. The numbers stored in memory for each frequency reflect the fact that the pulses are multiplied in amplitude by B+ when they are converted to the output waveform in the power stage. The scale factor (SF) likewise takes into account the variation in B+ compared with the assumed nominal value used to derive the stored data. Hence the multiplication factor $\frac{B+ ref}{B+}$. A value for B+ is acquired via an A/D converter as an input to the PWM microprocessor software.

More specifically and in the terms used in the software the values calculated for the pulse widths are as follows.

CONTROLLER CONFIGURATION AND 60° SYMMETRY

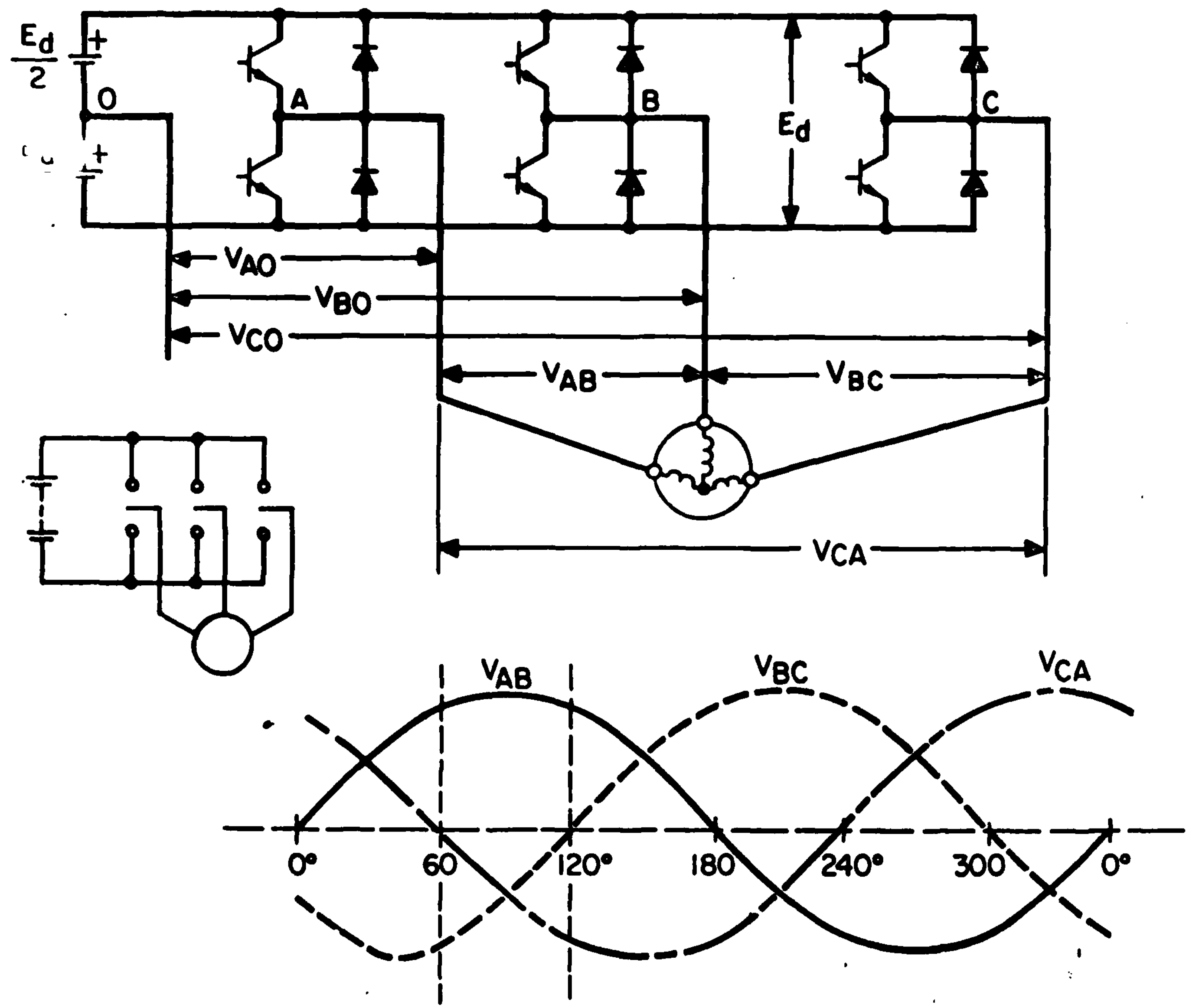
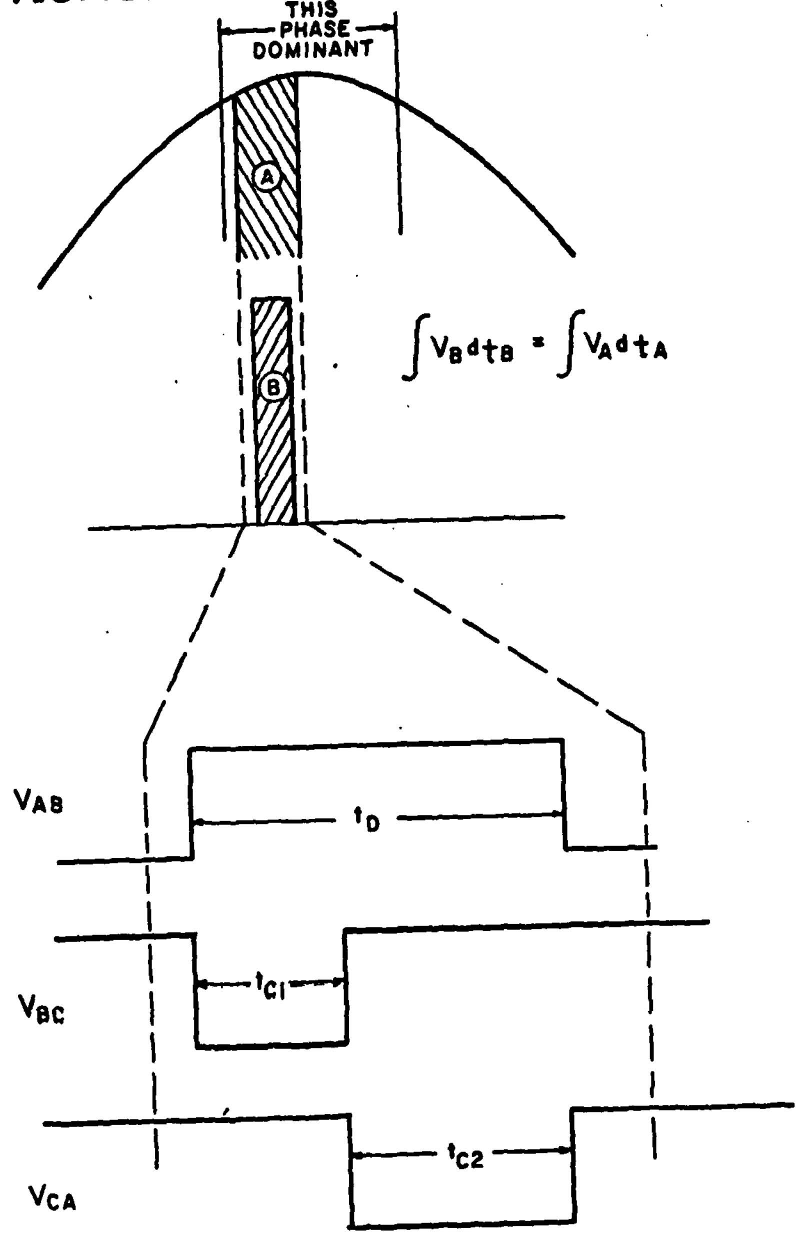
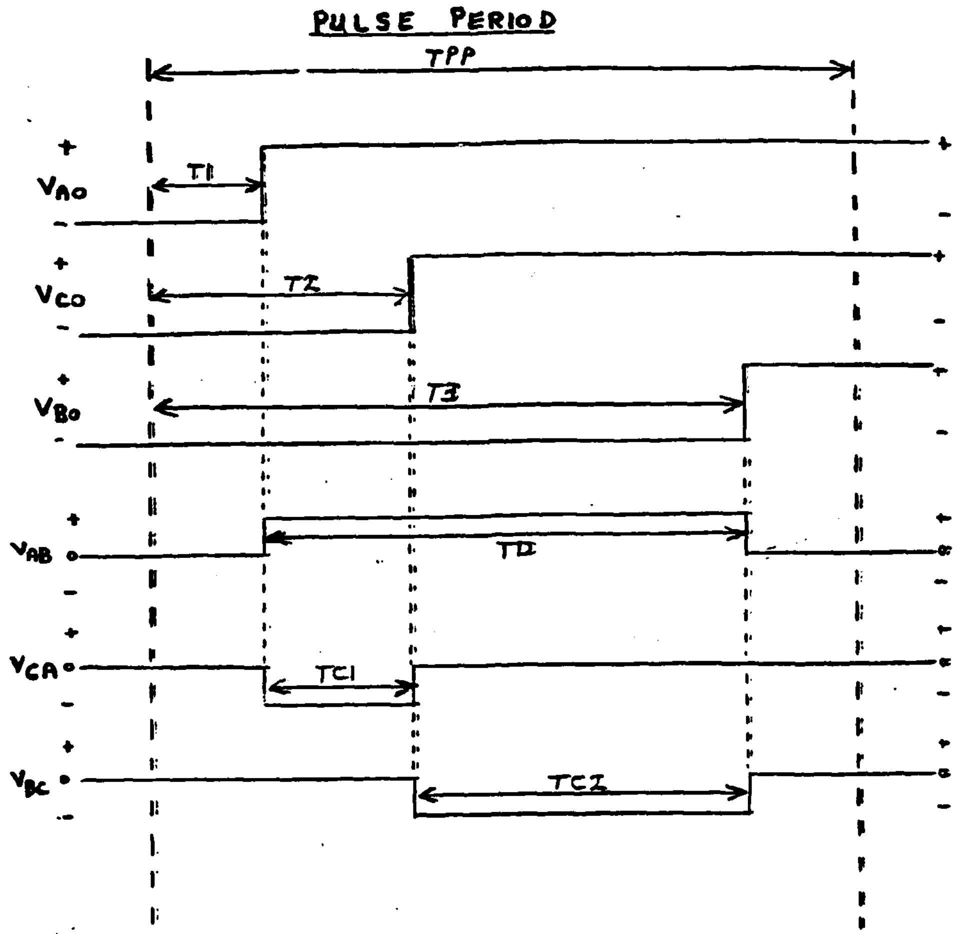


FIG. 2.1

MODULATION APPROACH



G.



- TD_{ref} + $TC2_{ref}$ STORED IN MEMORY

$$T_{ref} = \frac{1}{B^+_{ref}} \int_T^{T+TPP} V_{ref} \cdot \sin(\omega t + \phi)\, dt$$

- $SF = \frac{V_{OUT}}{V_{ref}} \cdot \frac{B^+_{ref}}{B^+}$

$TD = TD_{ref} \cdot SF$; $TC2 = TC2_{ref} \cdot SF$

- $T1 = \frac{TPP - TD}{2}$; $T3 = T1 + TD$

$T2 = T3 - TC2$ (or $T2 = T1 + TC2$)

SOFTWARE CONTROLLED TIMER OUTPUTS

FIG. 2.3

$$PWDom_n = \int_{\frac{\pi}{3}\left(1+\frac{n-1}{N}\right)}^{\frac{\pi}{3}\left(1+\frac{n}{N}\right)} \sin\omega t \cdot d\omega t \qquad PWCom_n = \int_{\frac{\pi}{3}\left(\frac{n-1}{N}\right)}^{\frac{\pi}{3}\left(\frac{n}{N}\right)} \sin\left(\omega t + \tfrac{2\pi}{3}\right) \cdot d\omega t$$

FOR UNIT AMPLITUDE REFERENCE, AND n=1 TO N. ωt IS THE ANGLE INTO THE CYCLE

Where N is number of pulse periods per 60° interval at the output frequency of interest F and N is the Nth pulse in the interval. The second complementary pulse widths are antisymmetrical with PWCOM and are not generated separately.

TOTAL PULSE PERIOD LENGTH	$T_{PP} = \frac{1}{F} \cdot \frac{1}{6} \cdot \frac{1}{N}$
PERIOD TO TIMER #1	$T_1 = (T_{PP} - PW_{DOM})/2$
PERIOD TO TIMER #3	$T3 = T_1 + PW_{DOM}$
PERIOD TO TIMER #2	$T2 = T3 - PW_{COM}$, $n = \text{ODD}$ $T2 = T1 + PW_{COM}$, $n = \text{EVEN}$

If V_{BC} is dominant in the 60° period of interest then the pole centerpoint voltages with respect to B- are

$$V_B = \overline{T_1} \cdot (n = \text{ODD}) + T3 \cdot (n = \text{EVEN})$$
$$V_A = \overline{T_2} \cdot (n = \text{ODD}) + T2 \cdot (n = \text{EVEN})$$
$$V_C = \overline{T_3} \cdot (n = \text{ODD}) + T1 \cdot (n = \text{EVEN})$$

There are two power switches in each pole, an upper and a lower. The base drive waveforms for upper switches (neglecting shoot through protection, etc.) have the same form as V_B, V_A, V_C. The drive for the lower switches the inverse of V_B, V_A, V_C.

The output line to line voltages are the physical combination of the pole voltages across the motor windings

$V_{BC} = V_B - V_C$

$V_{AB} = V_A - V_B$

$V_{CA} = V_C - V_A$

The PWM processor implements the control strategy of Fig. 2.4. A nominal V/F = constant curve is chosen such that the current in the unloaded motor is a minimum at each frequency. This corresponds to optimum excitation at that load at that frequency. In response to load change as sensed by the feedback processor, the voltage is increased to maintain the excitation at the optimum level although the load is changing. Demand for excessive voltage as sensed by monitoring the level of an error signal causes the controller to attempt to shed load by reducing frequency. If load reduces bringing the error signal within bounds the controller stops at the new frequency and will return to higher frequency if the load transient is removed. Alternatively it will continue on down to zero frequency - off - if the overload condition persists. Hysteresis incorporated into the level detection provides for stable operation in this mode.

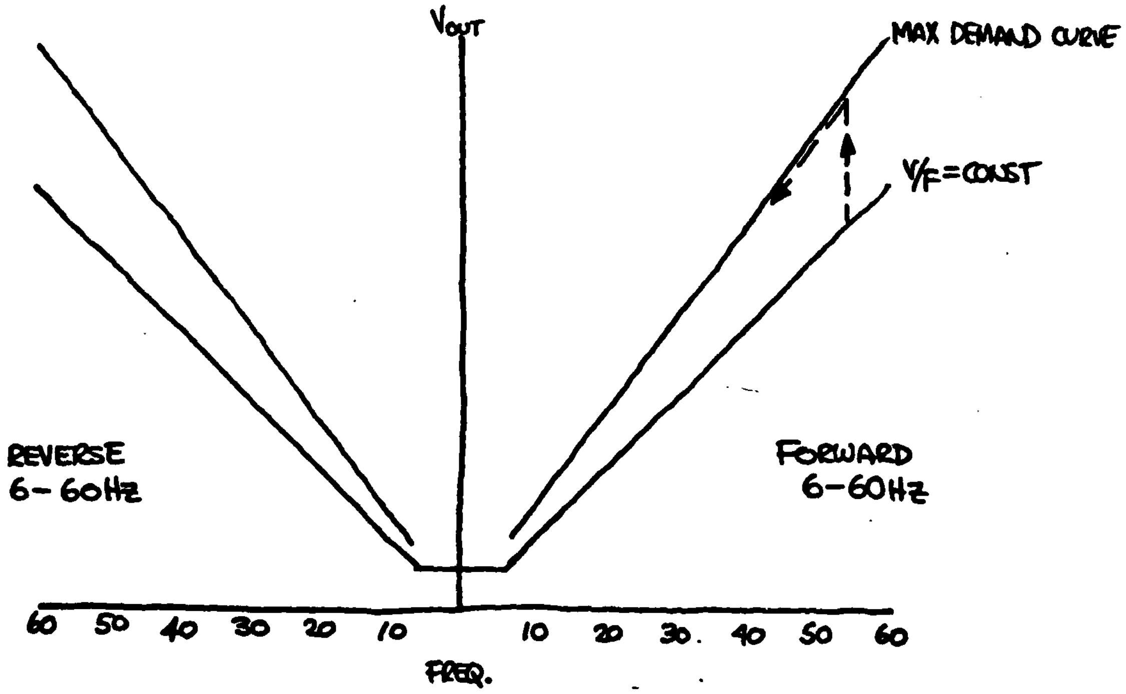


FIG. 2.4 CONTROL STRATEGY

3.0 Control System Design

3.1 Hardware

Design of the advanced motor controller was split into two basic sections: PWM Processor and Feedback Processor. An available laboratory power bridge was married with the processors. Figure 3.1 shows how each of these sections relates to the others. The power stage was employed in order to make operation with a test motor possible. The two processors are constructed on five wire-wrap boards, three being dedicated to the PWM processor, one to the feedback processor, and one to data acquisition and driver output functions.

Figure 3.2 and 3.3 show the construction of the PWM processor in block form. Analog speed command (terminal control optional) and battery B+ voltage is digitized and combined with feedback error information from the feedback processor to modify the output voltage scaling generated by the basic three phase sine wave algorithm. A monitor program to provide a terminal interface for optional keyboard control resides in PROM as well as the PWM routines. Figure 3.4 is a memory map showing locations in the PWM processor.

The configuration of the feedback processor is shown in Figure 3.5. The associated memory map appears in Figure 3.6. The feedback processor accepts the digitized form of the D-Q variables generated by the sensing and low pass filter circuits and computes the motor air gap flux using motor parameters stored in PROM. The calculated air gap flux is compared to a reference and the subsequent difference is digitally filtered and passed to the PWM processor through the V* (feedback error) latch.

Figure 3.7 is a hardware block diagram, showing the individual wire-wrap cards, their interconnections, and the System I/O. Appendix A contains schematics of the five cards and their interconnections.

3.2 Software

Figures 3.9 through 3.11 contain details of the PWM processor software. Figure 3.12 shows the feedback processor algorithm. Appendix B contains printouts of the system software.

The main control loop shown in Figure 3.9 first initializes all pointers, RAM, and flags. Next flags are set according to motor, power input, and command conditions. If no protection flags are set, the program ramps the output frequency up to the commanded frequency. Overload, B+, and reversing flags are checked each time through the loop. The PTM interrupt routine generates the dominant and complementary

times from frequency table information stored in PROM. The condition of the direction flag determines the order of output times and thus the output phase sequence. Prior to the time computations in the PTM interrupt routine, the input B+ voltage is read and the scale factor adjusted to keep the output independent of B+ variations. The quotient B+ ref/B+ is stored in a table in PROM for values from below B+ min to above B+ max. The feedback error, V*, is read during the beginning of each 60° output segment and is used to adjust the scale factor. During operation with a keyboard terminal, the monitor routine shown in Figure 3.11 is employe

The feedback microprocessor routine containing the flux computation and filtering algorithms is shown in Figure 3.12. The D-Q variables are digitized by four A/D converters after being generated as analog signals.

The air gap flux is then computed utilizing motor parameters stored as K1 in PROM. The calculated flux, Wx, is compared to the reference. Both K1 and reference R* are stored in PROM indexed to frequency. Thus, before calculating the flux and the error, the processor reads a word containing the output frequency from the PWM processor.

FIG. 3.1

ADVANCED MOTOR CONTROLLER

BLOCK DIAGRAM

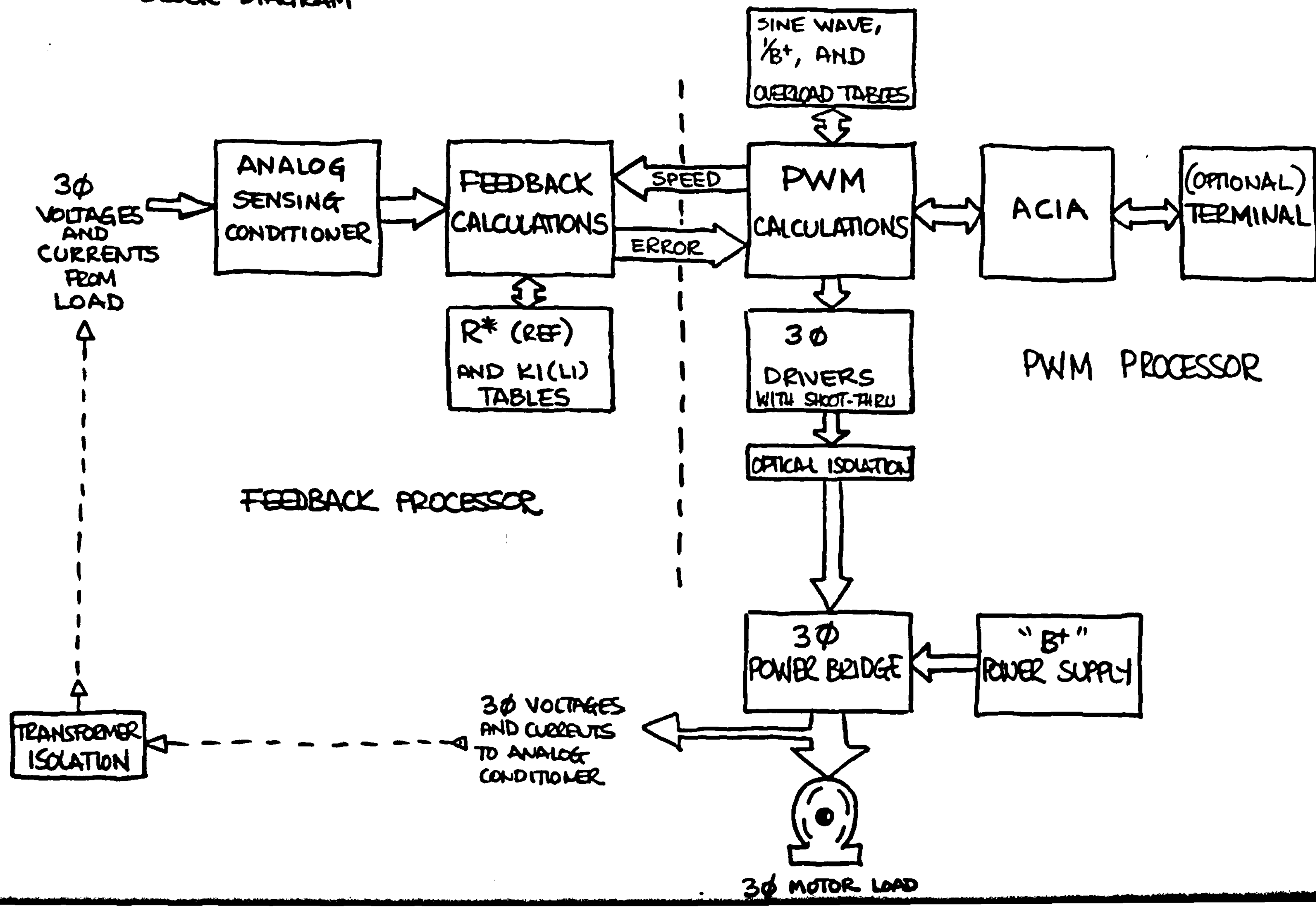


FIG. 3.2

PWM MICROPROCESSOR
BLOCK DIAGRAM

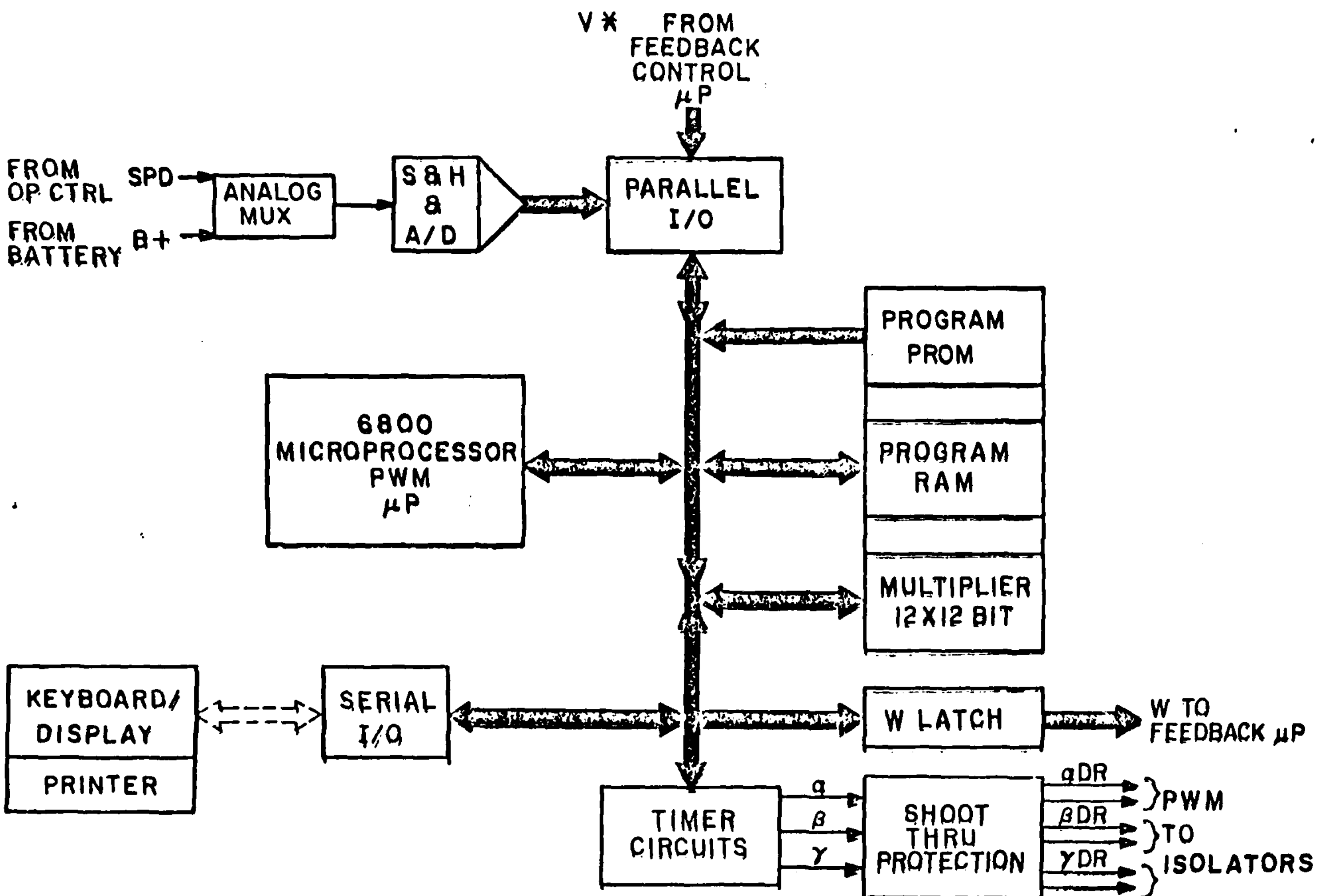
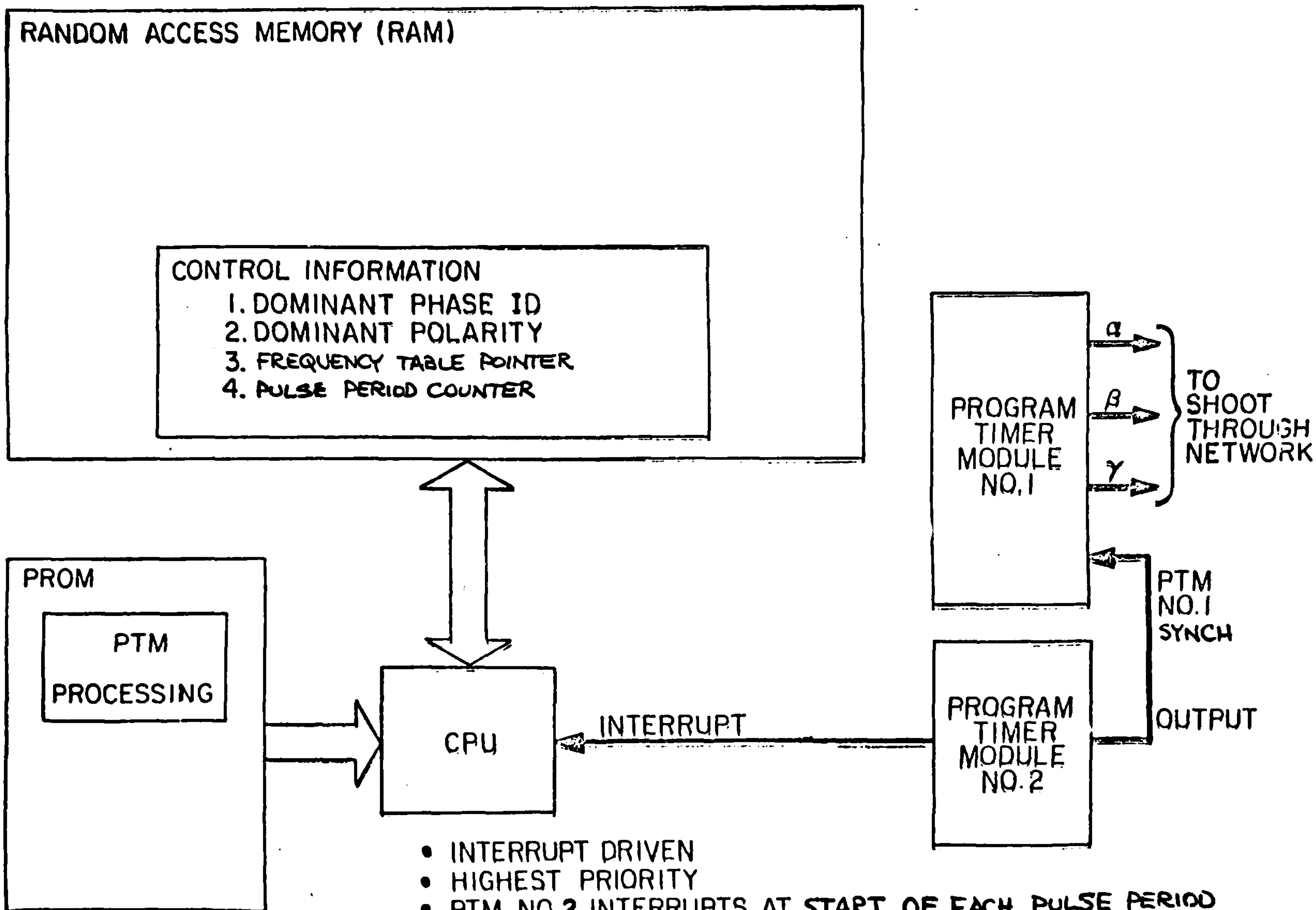


FIG. 3.3

TIMER CONTROLLER



- INTERRUPT DRIVEN
- HIGHEST PRIORITY
- PTM NO. 2 INTERRUPTS AT START OF EACH PULSE PERIOD

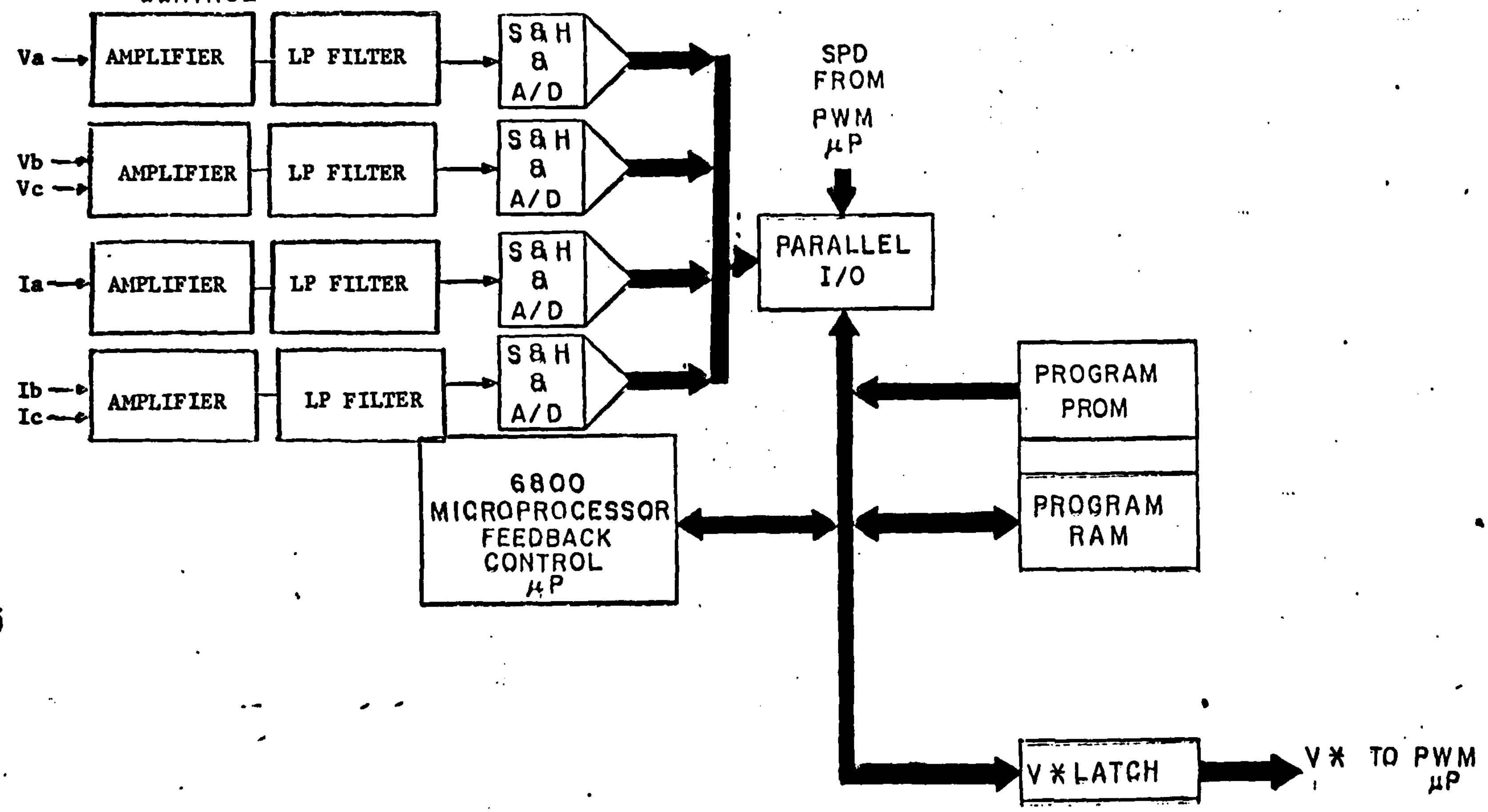
FIG. 3.4

HARDWARE MEMORY MAP
PWM PROCESSOR

SIZE		ADDRESS	FUNCTION
2 BYTES		FFFE, FFFF	RESET VECTOR
1024 X 8	RAM	FFFD FC00	MONITOR GLOBALS MCL GLOBALS
1024 X 8	RAM	FB00 F800	MONITOR GLOBALS
2048 X 8	PROM	F7FF F000	PMON9
2048 X8	PROM	EFFF E800	FREQUENCY TABLES
2048 X 8	PROM	E7FF E000	PTIS
2048 X 8	PROM	DFFF D800	MCL
1024 X 8	RAM	D7FF D400	MONITOR GLOBALS
	NOT USED	D3FF 7000	
4096 X 8	ACIA	6FFF 6000	ACIA ADDRESSING
4096 X 8	PIA	5FFF 5000	PIA ADDRESSING
4096 X 8	PTM1	4FFF 4000	PTM1 ADDRESSING
4096 X 8	PTM2	3FFF 3000	PTM2 ADDRESSING
4096 X 8	MULTIPLIER 'W' LATCH	2FFF 2000	MULTIPLIER ADDRESSING FREQUENCY INFORMATION ADDRESSABLE LATCH
4096 X 8	NOT USED	1FFF 1000	
4096 X 8	NOT USED	0FFF 0400	
1024 X 8	RAM	03FF 0000	MCL GLOBALS & TEMPORARY STORAGE

FIG. 3.5

FEEDBACK CONTROL MICROPROCESSOR BLOCK DIAGRAM



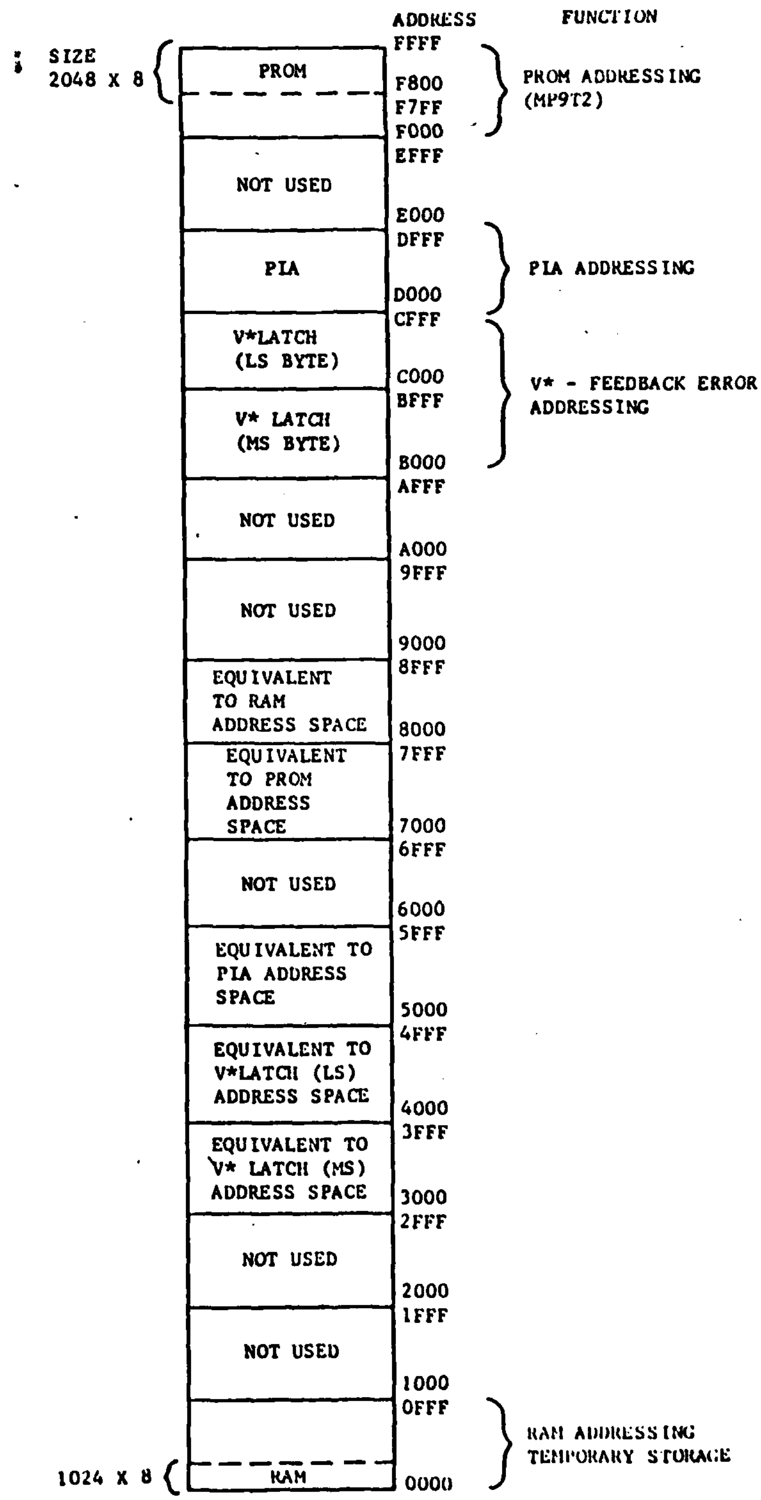


FIG. 3.6

HARDWARE MEMORY MAP
FEEDBACK PROCESSOR

FIG 3.7

ADVANCED MOTOR CONTROLLER

FUNCTIONAL WIRING DIAGRAM, CHASSIS

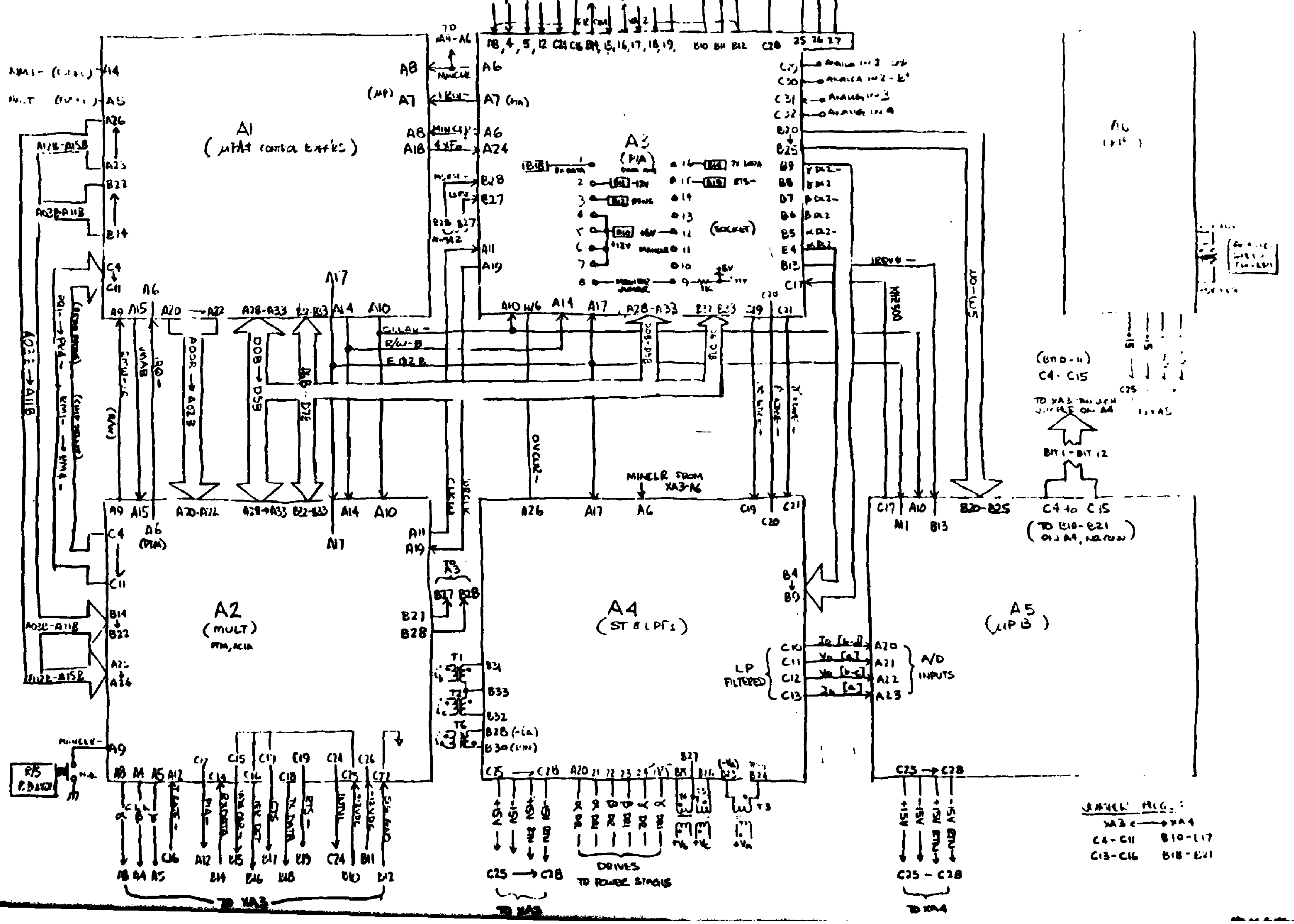
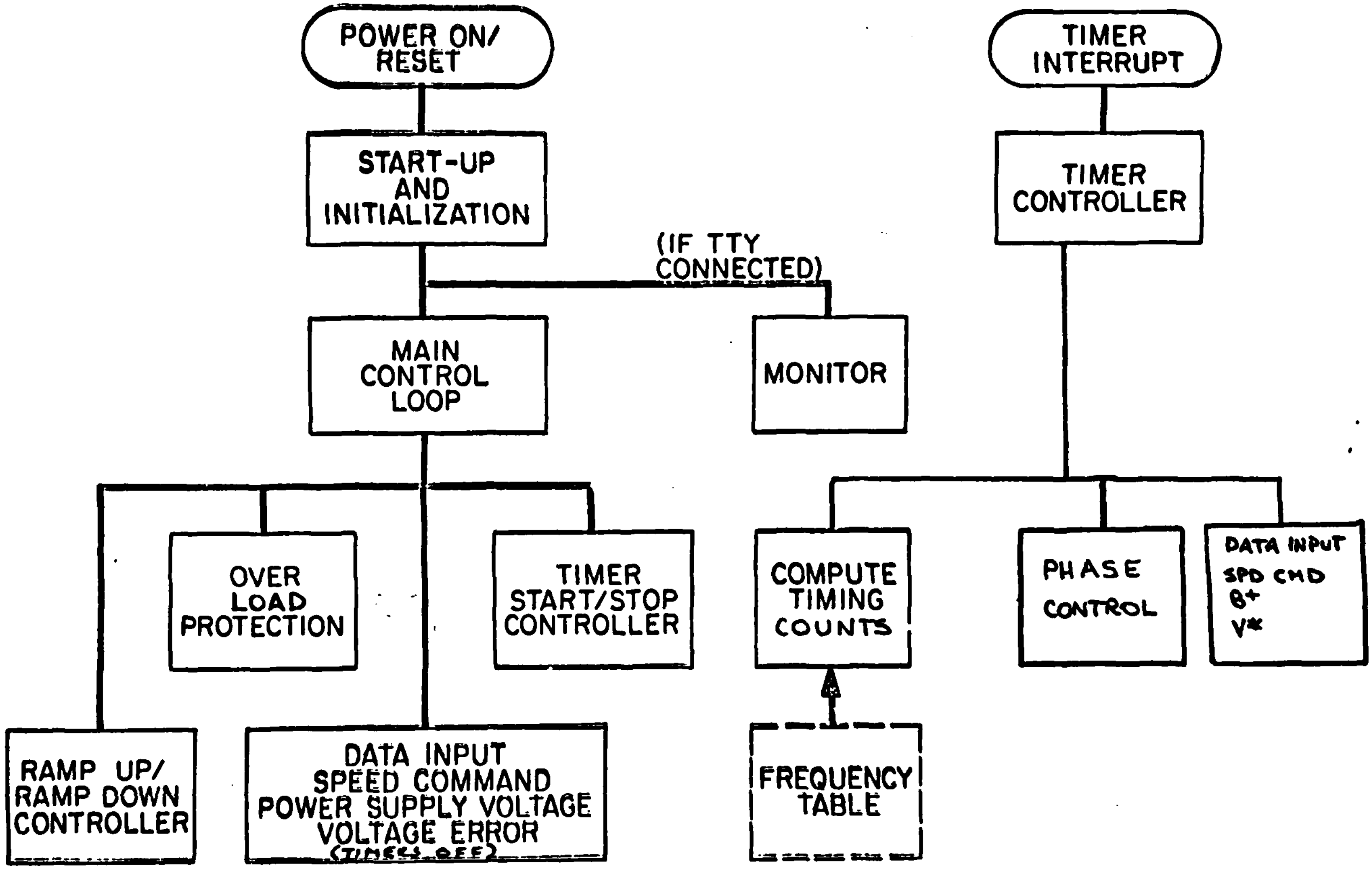


FIG. 3.8
PWM PROCESSOR SOFTWARE



- TWO CONCURRENT TASKS
- INTERFACE THROUGH CONTROL PACKET

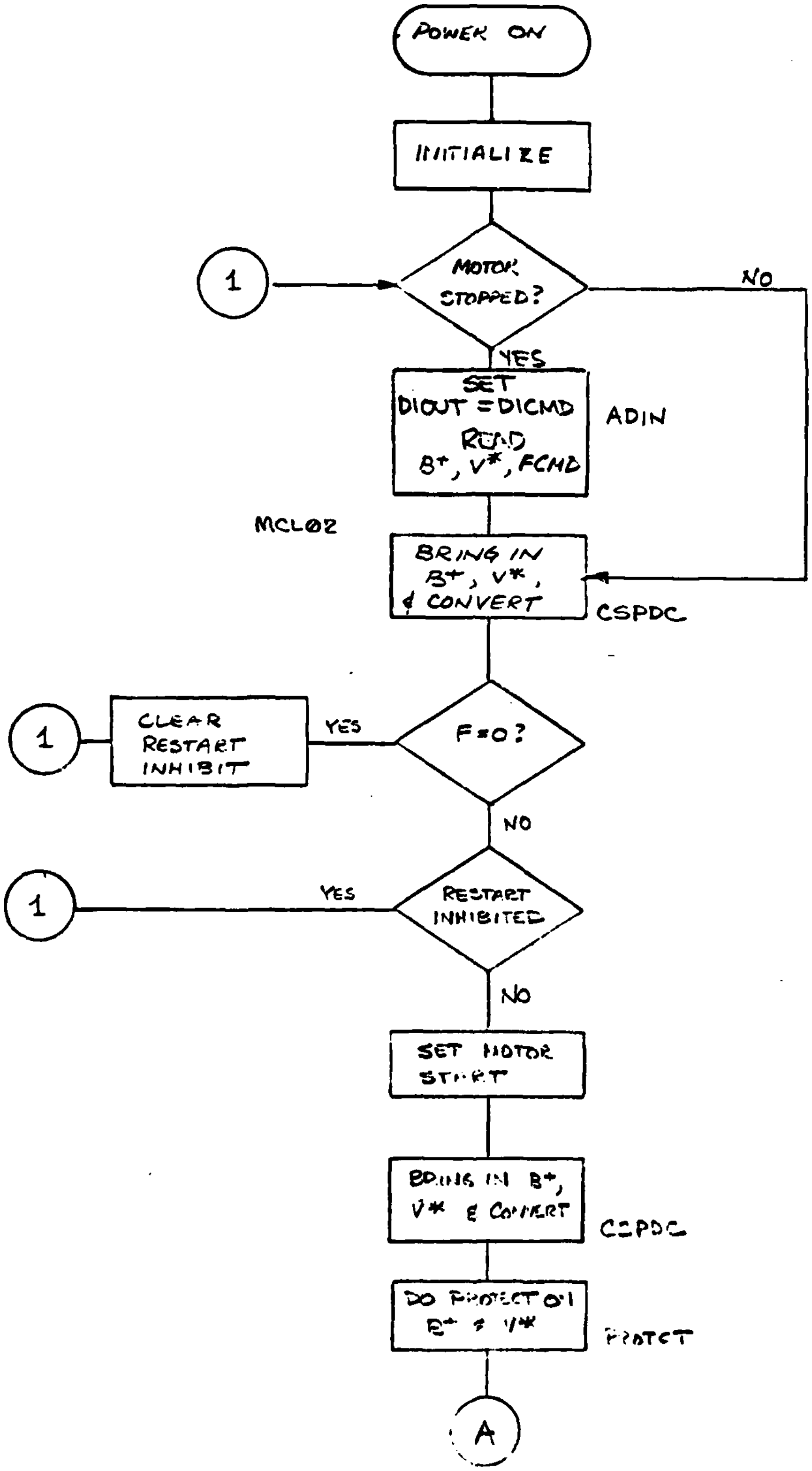


FIG 3.9 MAIN CONTROL LOOP ALGORITHM (SHEET 1)

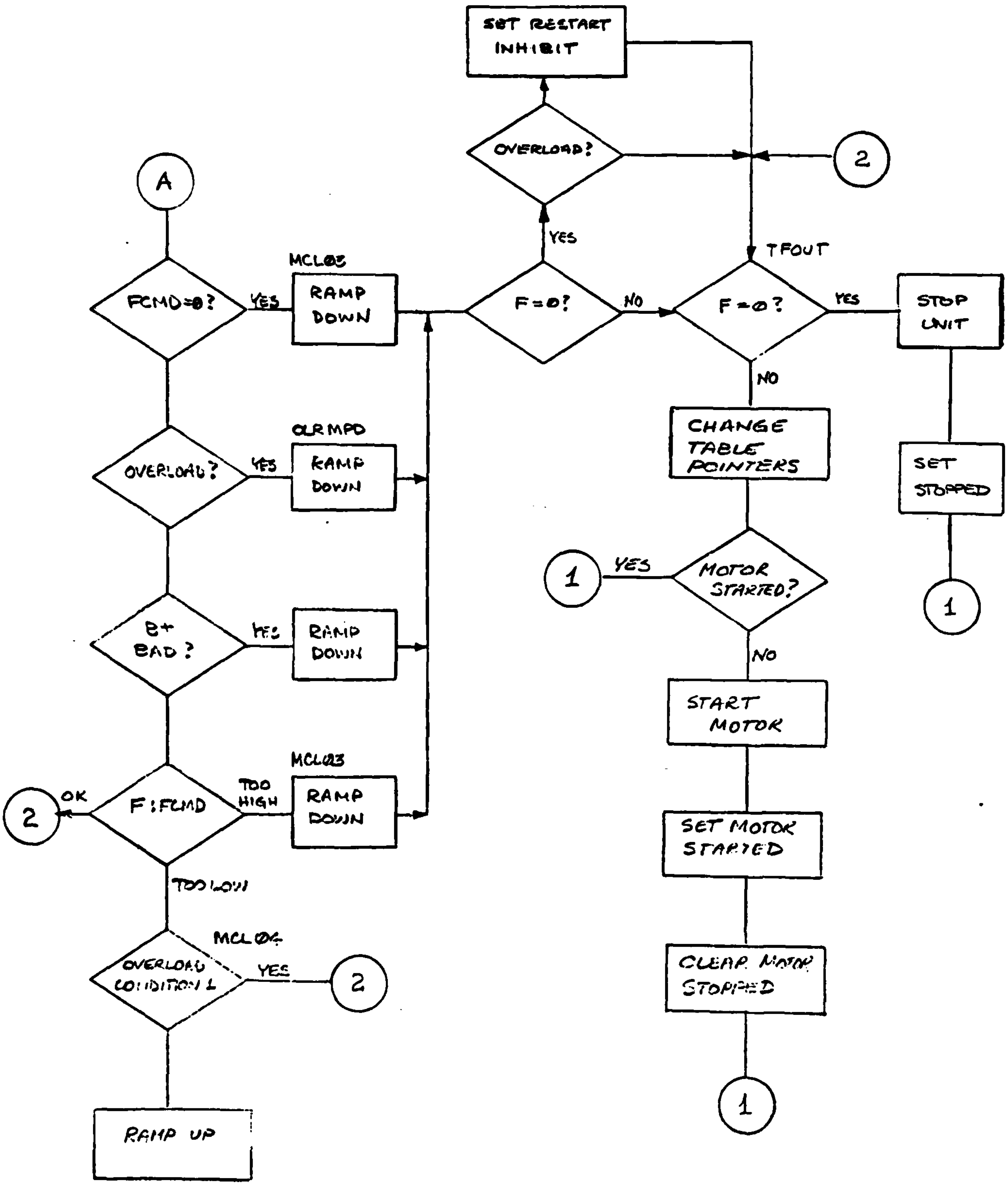


FIG 3.9 MAIN CONTROL LOOP ALGORITHM (SHEET 2)

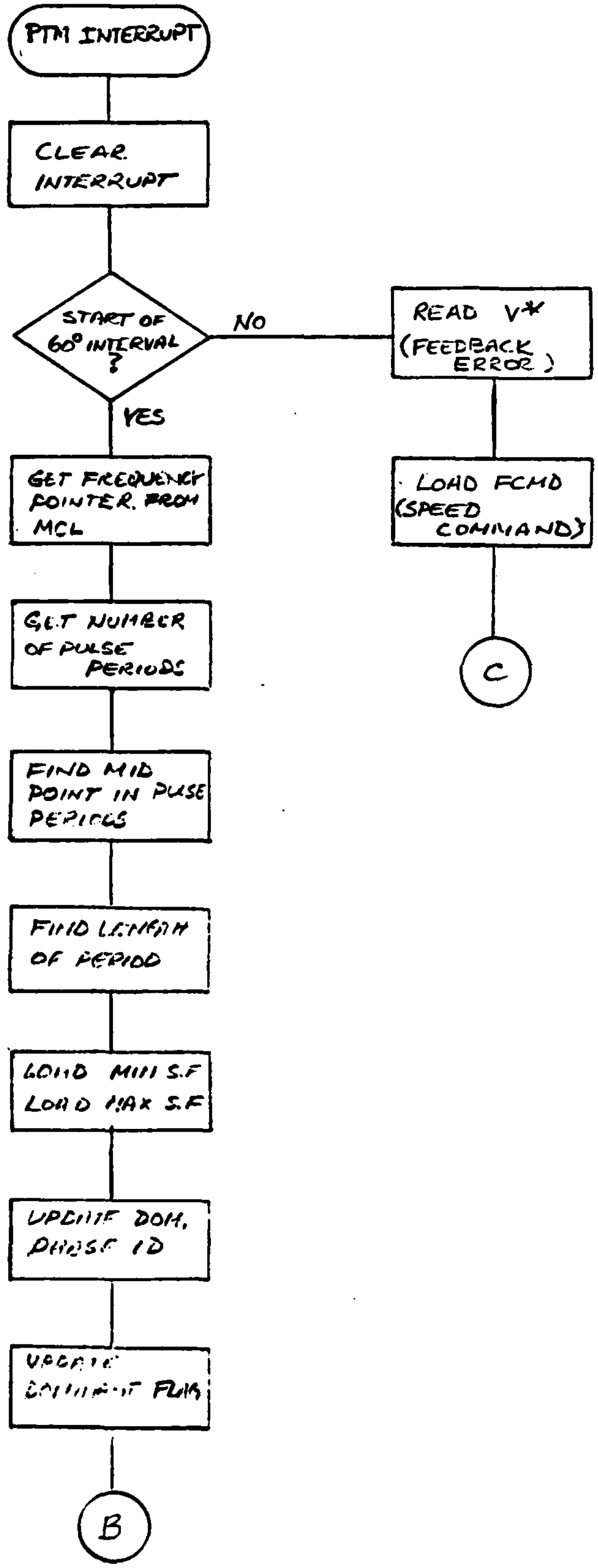
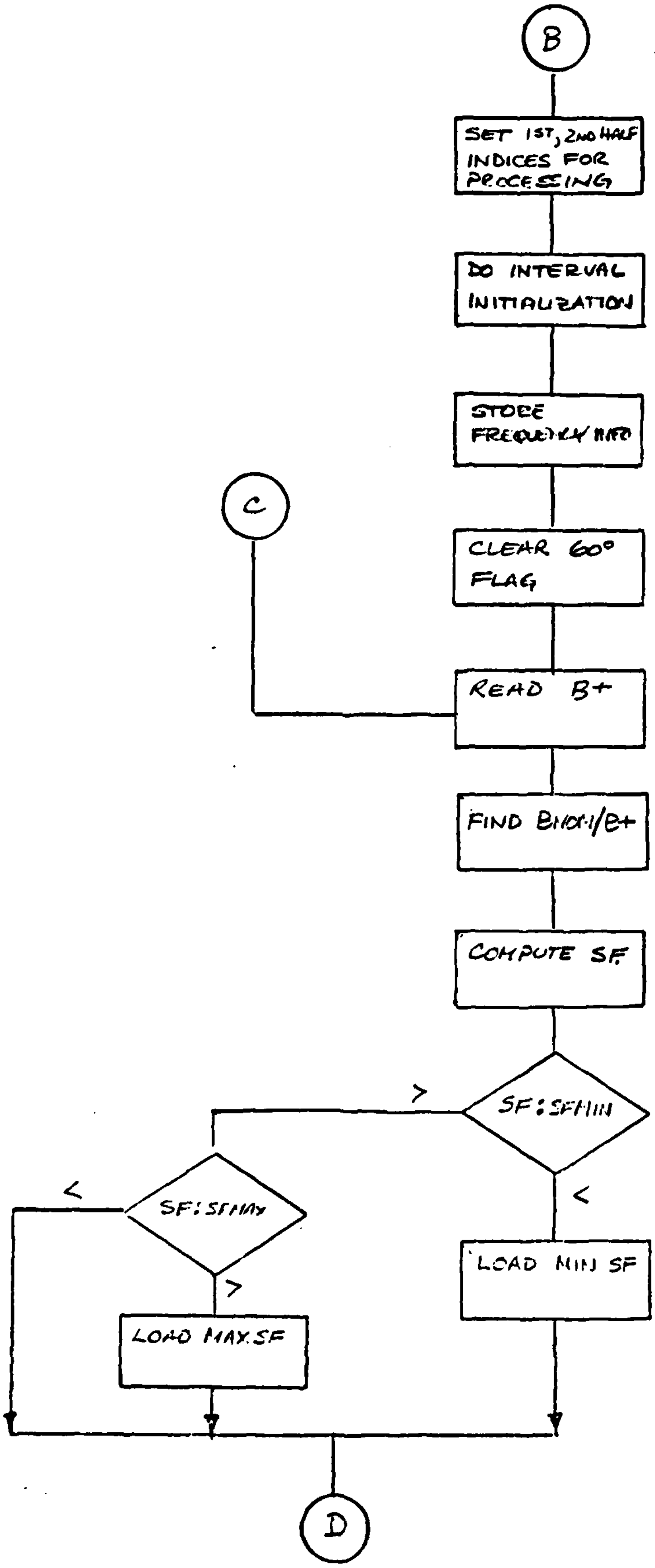
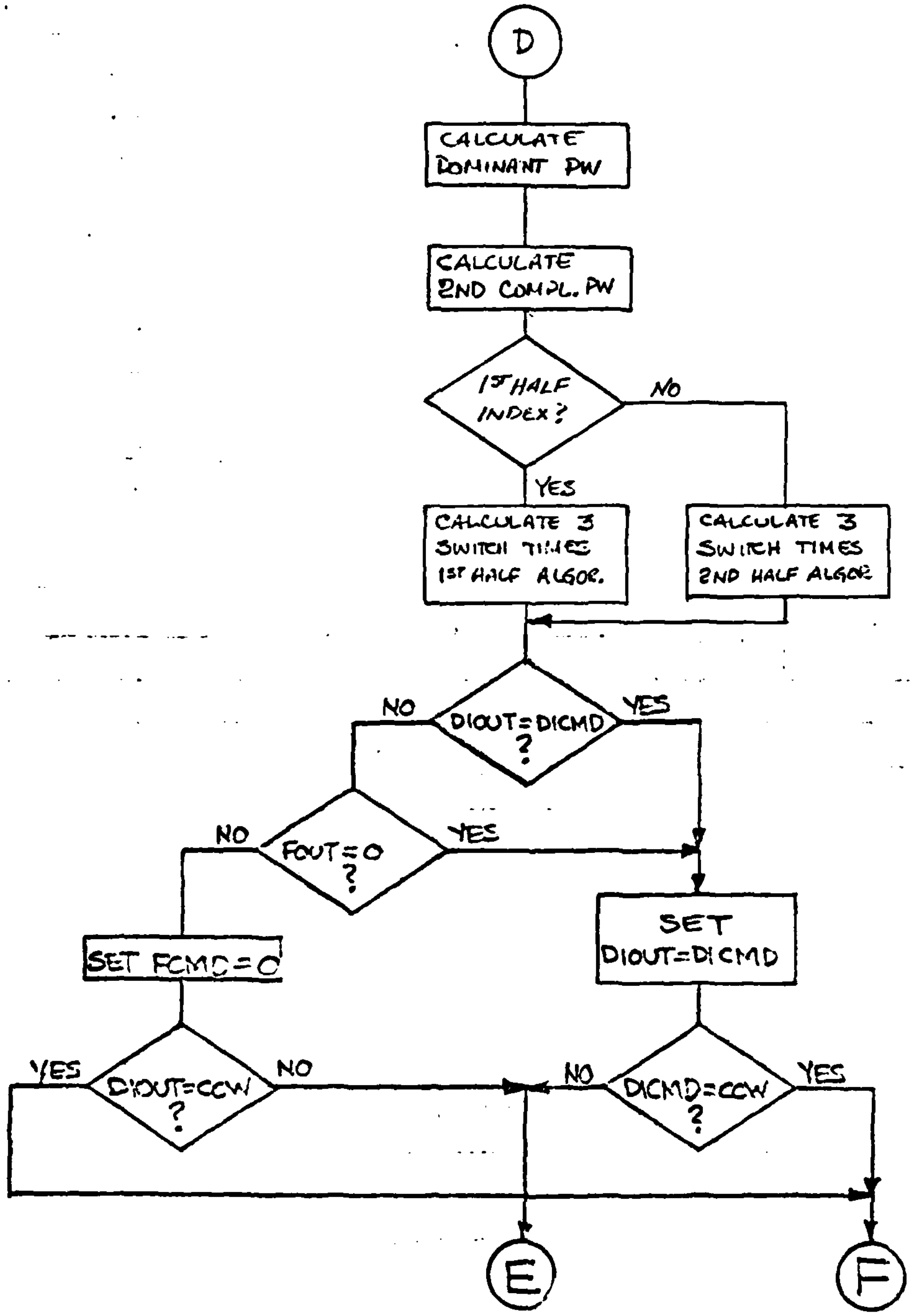
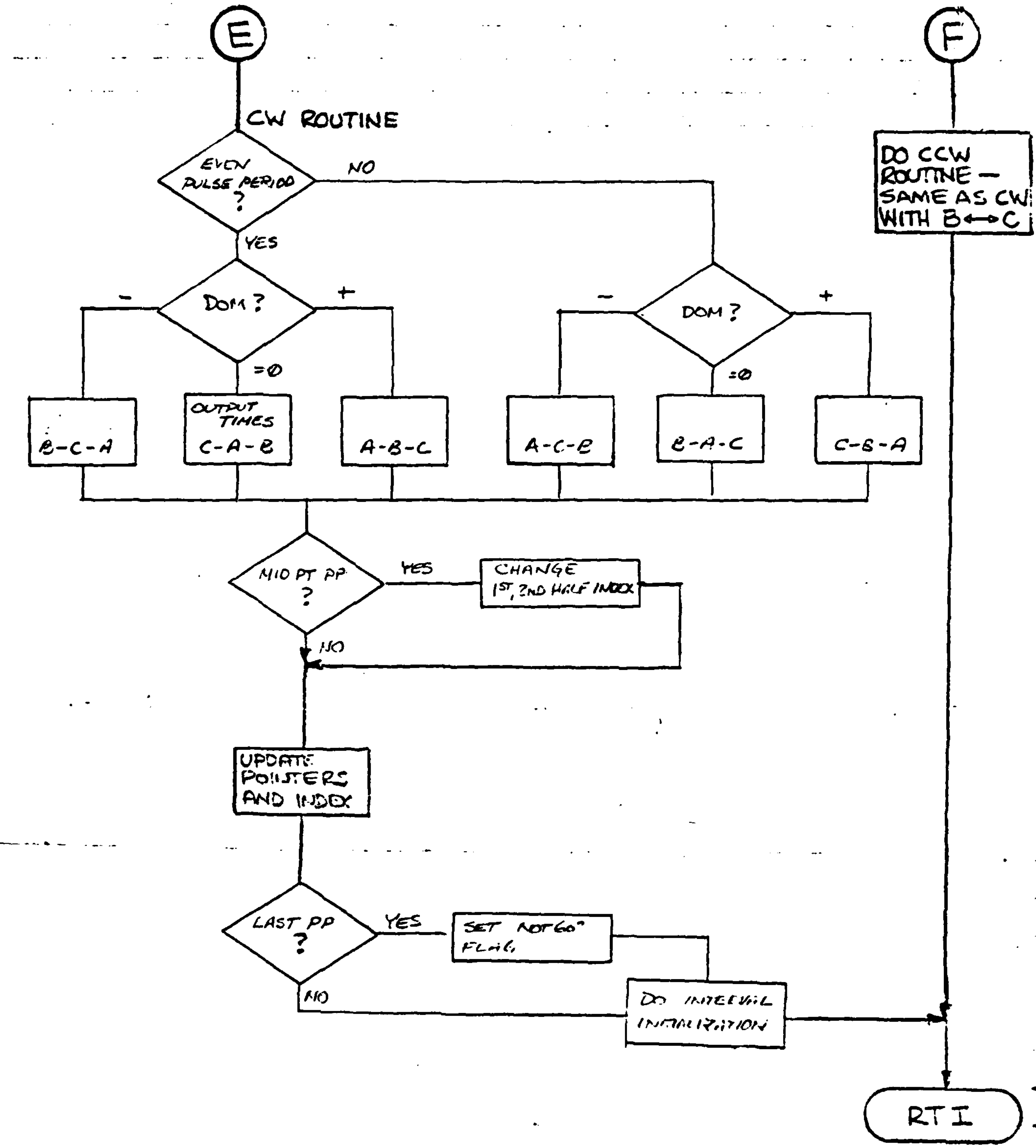


FIG 3.10 PTM INTERRUPT ROUTINE (SHEET 1)


B
SET 1ST, 2ND HALF INDICES FOR PROCESSING
DO INTERVAL INITIALIZATION
STORE FREQUENCY INFO
C
CLEAR 60° FLAG
READ B+
FIND BNOM/B+
COMPUTE SF
SF : SFMIN
>
<
SF : SFMAX
<
>
LOAD MAX SF
LOAD MIN SF
D


D
CALCULATE DOMINANT PW
CALCULATE 2ND COMPL. PW
1ST HALF INDEX?
NO
YES
CALCULATE 3 SWITCH TIMES 1ST HALF ALGOR.
CALCULATE 3 SWITCH TIMES 2ND HALF ALGOR.
NO
DIOUT=DICMD ?
YES
NO
FOUT=0 ?
YES
SET FCMD=0
SET DIOUT=DICMD
YES
DIOUT=CCW ?
NO
NO
DICMD=CCW ?
YES
E
F


E
F
CW ROUTINE
EVEN PULSE PERIOD ?
NO
YES
DOM ?
-
+
=Ø
B-C-A
OUTPUT TIMES C-A-B
A-B-C
A-C-B
B-A-C
C-B-A
DO CCW ROUTINE — SAME AS CW WITH B↔C
MID PT PP ?
YES
CHANGE 1ST, 2ND HALF INDEX
NO
UPDATE POINTERS AND INDEX
LAST PP ?
YES
SET NOT GO FLAG
NO
DO INTERVAL INITIALIZATION
RTI

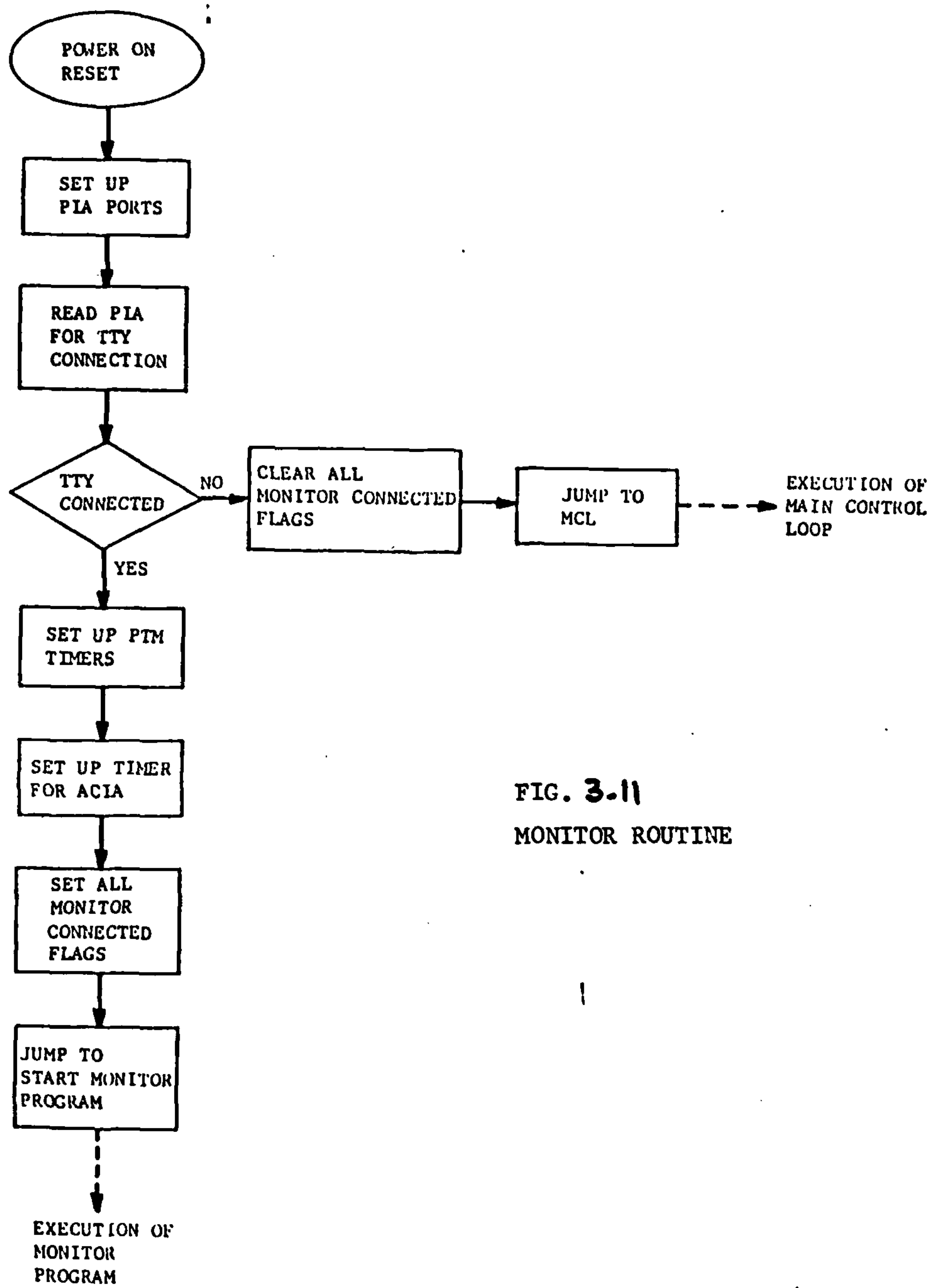


FIG. 3.11
MONITOR ROUTINE

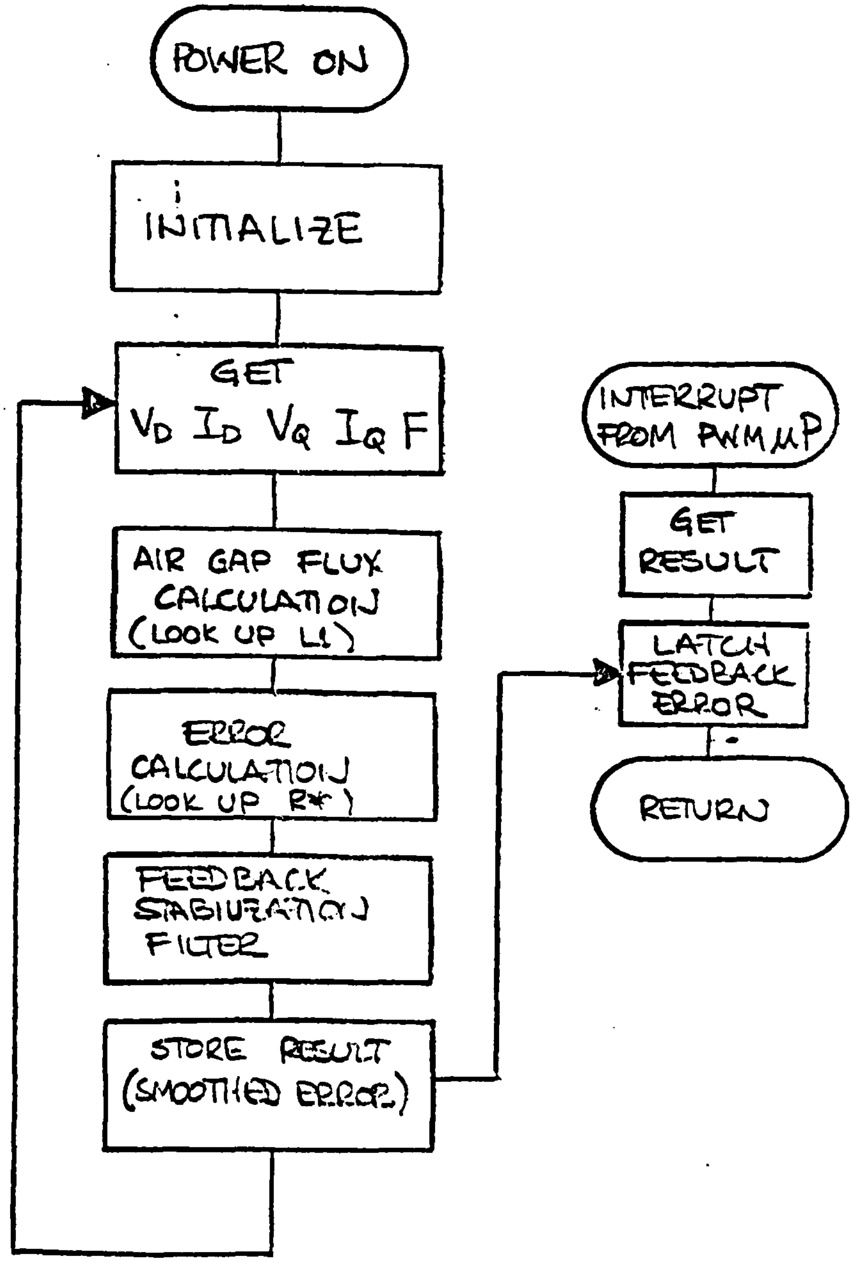


FIG. 3.12
FEEDBACK MICROPROCESSOR
ALGORITHM

4.0 Test Measurements

4.1 Flux and Control Loops

To establish an efficient operating curve, the motor was excited at frequencies from six to sixty hertz while varying the excitation voltage and observing the input current. A plot of the points of voltage required for minimum current was then made against frequency. As expected, this curve was a straight line (v/f = constant). The table used by the PWM processor to determine the open-loop operating point was generated using the linear v/f information.

The system operated properly under open loop conditions, responding to speed and amplitude commands from the keyboard.

Next, phase voltages and currents were sensed and the D-Q variables successfully generated, converted, and used in the flux calculation. Figure 4.2D shows the direct (bottom) and quadrature (top) currents as they exist at the analog input to their A/D converters. The D-Q voltage waveforms are similar. As desired, the amplitudes of the two components were the same and their phase differed by 90 degrees. Ø, the angle between V_D and I_D, was seen to vary with the load from 20 to 62 degrees. This range of phase angle is due to the choice of an operating point near minimum excitation current. Figure 4.1 schematically shows how the D-Q variables are generated in the system.

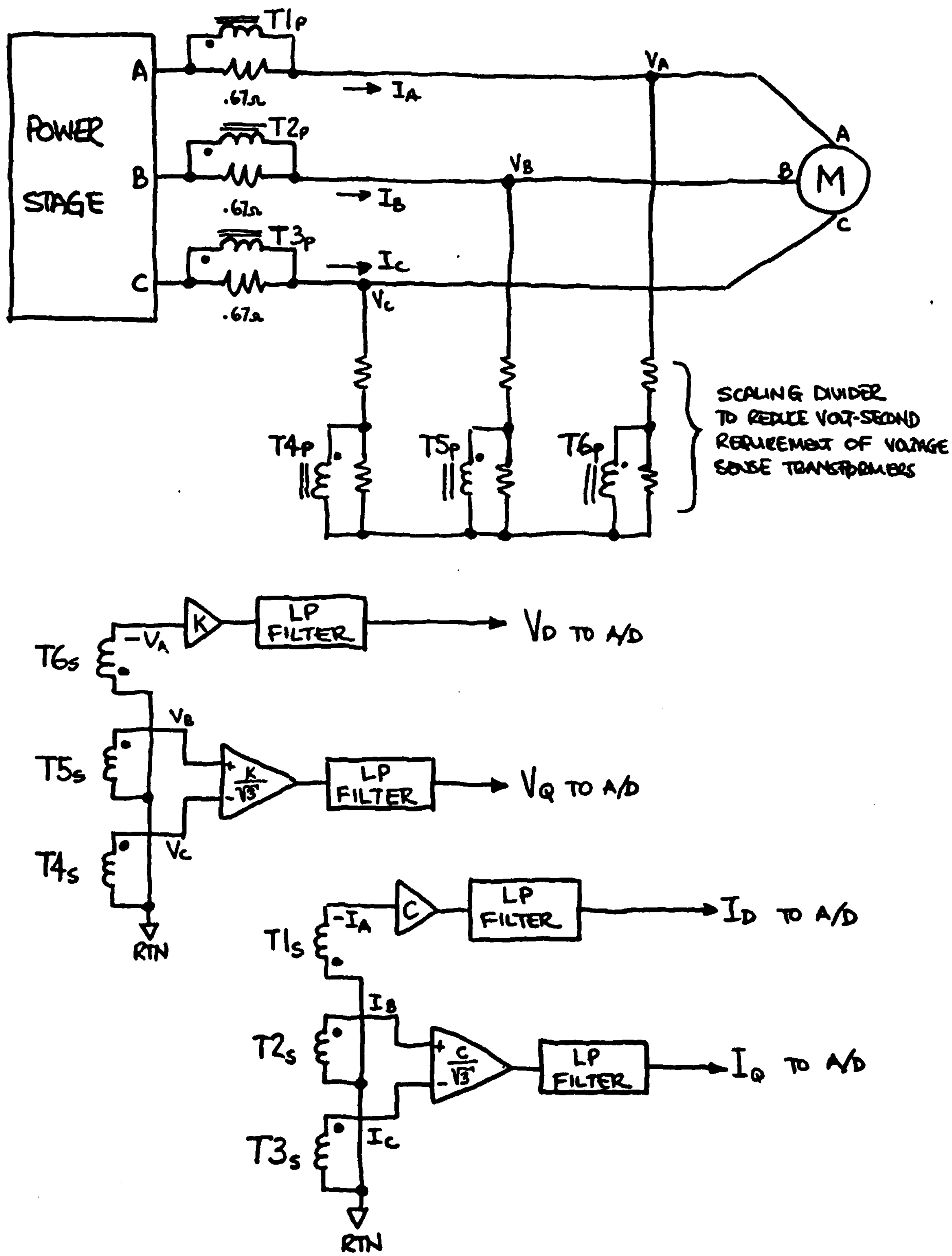
Four dedicated A/D converters digitize the analog D-Q variables. The A/D converters are sampled at a high rate compared to the output frequency, so their values may be considered as being taken simultaneously for purposes of the flux calculations.

An estimate of parameters of the test motor was made and simultation using an H.P. 9830 computer was performed to confirm that the parameters chosen would yield values of current and flux in keeping with the test motor size. These parameters were then used in the generation of the various tables required for flux calculation.

Operation in the open-loop mode yielded values of calculated flux consistent with the simulation. Table 4.1 below gives a set of measurements and calculations performed by the sense circuits and the feedback processor. A Hewlett-Packard 9611 was used to monitor operation of the microprocessor.

In closed loop operation, the calculated value of air gap flux, Wx, is subtracted from a fixed reference value, R*, and the resulting error signal is digitally filtered and transmitted to the PWM processor. The filter is used to smooth the result of the Wx calculation and to provide a controlled pole in the loop response of the system. The filtered error signal is used by the PWM processor to modify the output amplitude so as to reduce the error and keep the calculated value of Wx constant. In open loop operation, Wx values were observed by setting the reference, R*, to zero and observing the smoothed error.

FIG. 4.1 D-Q VARIABLE GENERATION



WORK SHEET
WESTINGHOUSE FORM 8295 E

S-2 31
DB2

INSTANTANEOUS READINGS	I_Q MEAS HEX	I_Q MEAS DEC	V_D MEAS H	V_D MEAS D	V_Q MEAS H	V_Q MEAS D	I_D MEAS H	I_D MEAS D	$V_D I_Q$ CALC	$V_Q I_D$ CALC	W_X "RAW"	"RAW" MEANS THAT $I^2 X_{L1}$ NOT SUBTRACTED.		
1	FC	-.15A	40	64V	0D	13V	09	.45A	-9.75	5.85	15.6	APPROX. CONSTANT AVG = 17.8		
2	04	.29	29	41	38	56	0A	.50	11.89	28	16.11			
3	09	.45	E8	-24	33	51	02	.10	-10.8	5.1	15.9			
4	02	.10	32	50	32	50	0A	.50	5	25	20			
5	FE	-.10	3E	62	1D	29	0A	.50	-6.2	14.5	20.7			
6	FA	-.30	3F	63	00	0	0A	.50	-18.9	0	18.9			
	CALCULATE		$\sqrt{I_D^2 + I_Q^2}$		FOR	ABOVE	CASES:	1	.474	$\times 14.2 (X_{L1})$		= 6.36 →	W_X (RAW − I^2X) 8.96	AVG = 10.6
								2	.578			8.09	8.02	
								3	.461			6.45	9.45	
								4	.510			7.14	12.86	
								5	.510			7.14	13.56	
[ADR 10 → 13]								6	.583			8.16	10.74	
P2-P4 (HEX)	0165	0189	0175											
P2-P4 (DEC)	11.15	12.28	11.65											

THESE VALUES LOOK GOOD (AVG = 11.69), ESPECIALLY CONSIDERING THEY ARE AT THE LOW END OF THE SCALE

E* AT μPA/B INTERFACE (H) 0315 (D) 24.6
E* BEFORE ↓ ↓ 06A0 (D) 13.25

V/X AT OUTPUT OF ADDER ROUTINE: ADR 32 AND 33 (4) 016X [AVG]

P2-P4 SIG	...	64 → 8	4 → .5	.25 → .03	
0165	→	0001	0110	0101	= 11.15
0189	→	0001	1000	1001	= 12.28
0175	→	0001	0111	0101	= 11.65

Since the measurements of Wx were in accord with expectations, a reference level close to the value of Wx at no load was inserted (instead of zero) and an attempt at closed loop operation was made. The motor/controller system was successfully exercised in the closed loop mode. Under control of flux loop calculations made by the feedback processor, the controller increased the output voltage amplitude to compensate for losses when the motor was loaded and decreased the output when the load was reduced, keeping the calculated value of flux constant. The "feel" of loading the motor by hand was dramatically stiffer in this mode. Stepping up to and down from the selected frequency under closed loop conditions also functioned properly.

4.2 Spectrum of PWM Output Motor Load

A Federal Scientific model UA-500 (ubiquitious) spectrum analyzer was used to examine the voltage and current frequency spectrum of the motor controller power stage output while driving the three phase test motor. Figures 4.3 through 4.5 are photographs of the output waveforms and spectra, taken under various conditions.

Notice that the amplitudes of the unwanted frequencies in the current spectrum stay constant as the fundamental rises with increased load. Observe also that very little unwanted output energy is evident below 1 KHz under any operating circumstance. The ratio of the amplitude of output current at the fundamental frequency to that of the next highest component, about 1.3 KHz, with the motor loaded to one-fifth horsepower is in excess of 33db.

Figures 4.2 through 4.5 also show output voltages and currents at various output frequencies. The current ripple stays approximately constant in amplitude over the output frequency range due to the inverse relation between the PWM carrier frequency and the output frequency. Comparison of the unloaded currents with the corresponding loaded output currents, as in figures 4.4B and D and 4.5A and C, shows that the current ripple remains constant as the output current increases. This phenomenon is responsible for the spectral purity observed in the loaded output current photographs.

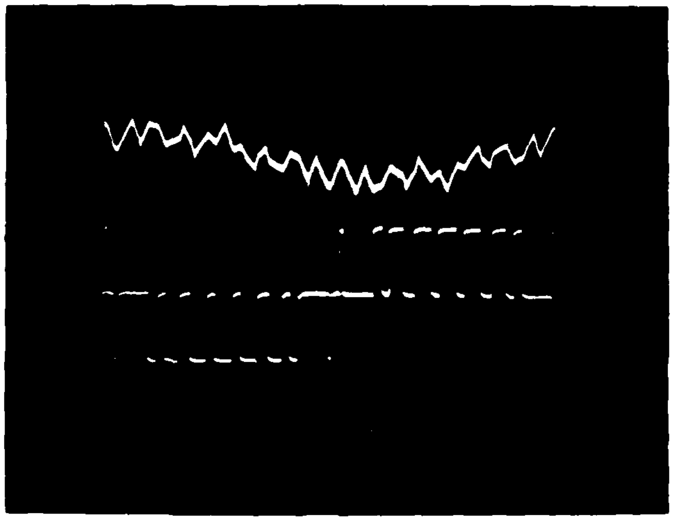
4.3 Reversing

A routine to step down to zero speed and reverse the output direction in response to the setting of a direction flag from the keyboard has been incorporated into the software and successfully tested. It remains to make some minor modifications to the flux loop D-Q inputs to properly calculate the Wx value in the reverse mode. This modification, as well as the reading of speed and direction inputs from an analog source (potentiometer), will be accomplished next.

4.4 Overload Protection

The software needed to sense overload at a preset level above the chosen operating point, to ramp down to an acceptable level, and to ramp back up when the overload is relieved has been incorporated in the control program.

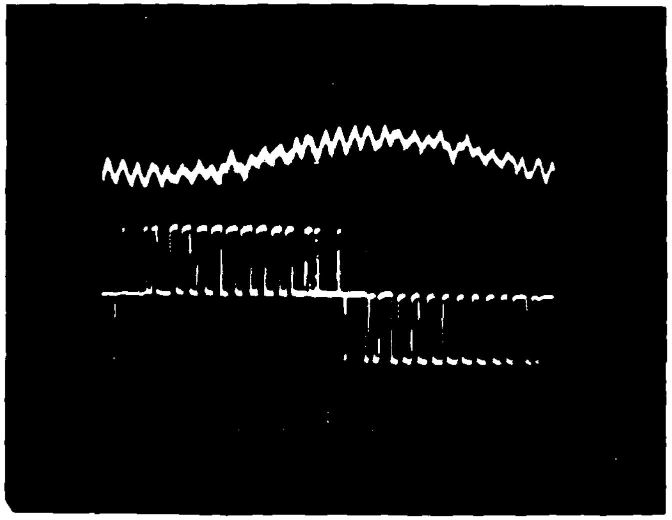
This feature will not be tested until the proposed Phase III of the program when the power stage is added and a total drive system tested.



OUTPUT CURRENT 1A/DIV

L-L OUTPUT VOLTAGE 100V/DIV

A. 60Hz, NL

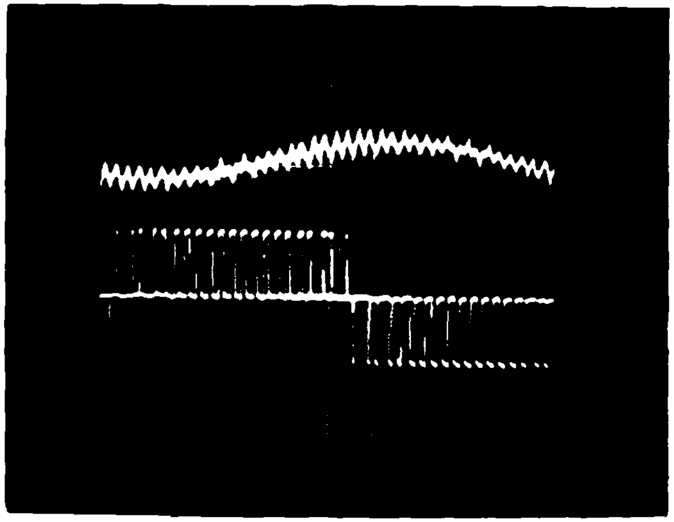


OUTPUT CURRENT 1A/DIV

LL-OUTPUT VOLTAGE 100V/DIV

C. 55Hz, NL

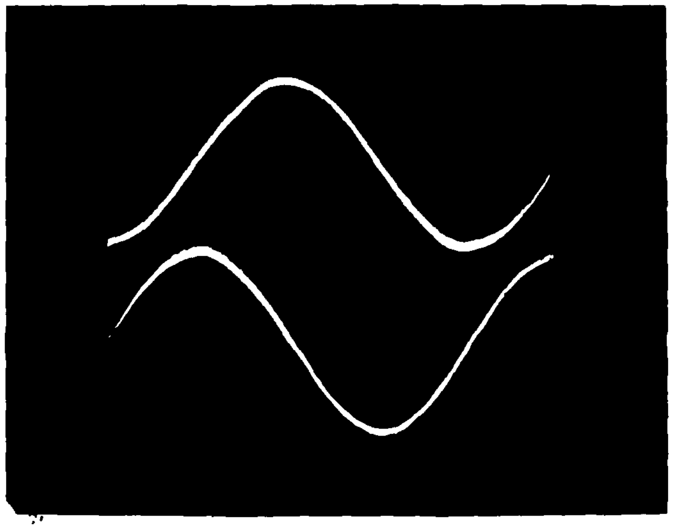
FIGURE 4.2



OUTPUT CURRENT 1A/DIV

L-L OUTPUT VOLTAGE 100V/DIV

B. 41 Hz, NL



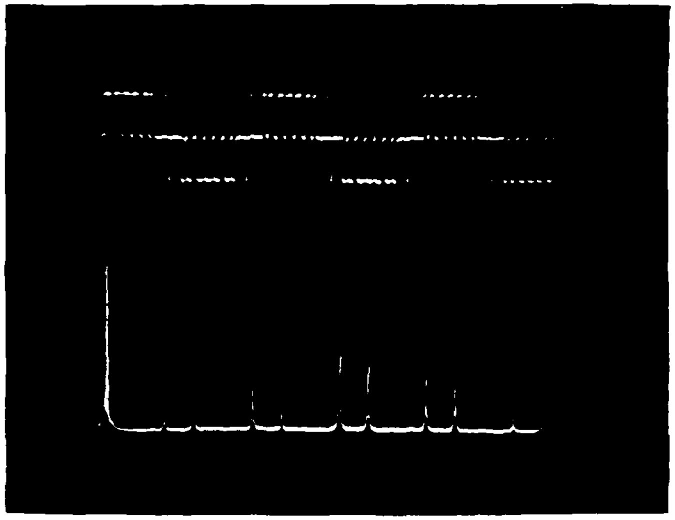
I_Q (FILTERED) .2V/DIV

I_D (FILTERED) .2V/DIV

TIME, 2ms/DIV

[SCALE FACTOR 1V=1.25A]

D. 60Hz, NL

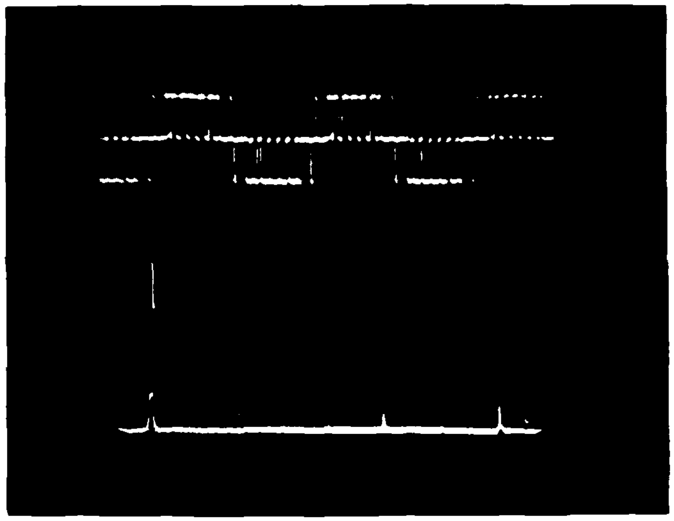


OUTPUT PWM VOLTAGE

VOLTAGE SPECTRUM 200Hz/DIV REL AMP =1/DIV

OUTPUT PWM VOLTAGE

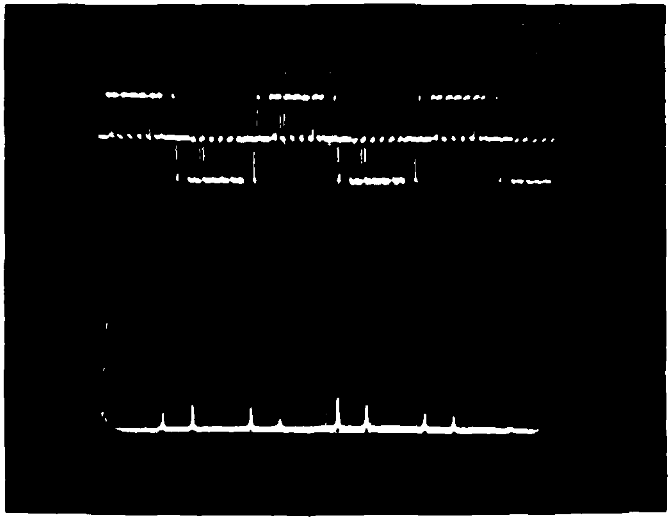
VOLTAGE SPECTRUM 50Hz/DIV REL AMP =1/DIV



A. 60Hz OUTPUT, NO LOAD ON MOTOR (NL)
18 PULSES/OUTPUT CYCLE

B. 60Hz OUTPUT, NL

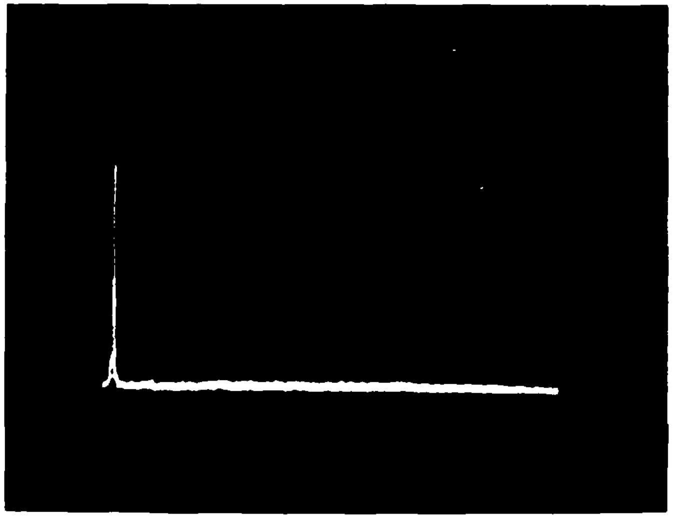
C. 60Hz, NL



OUTPUT PWM VOLTAGE

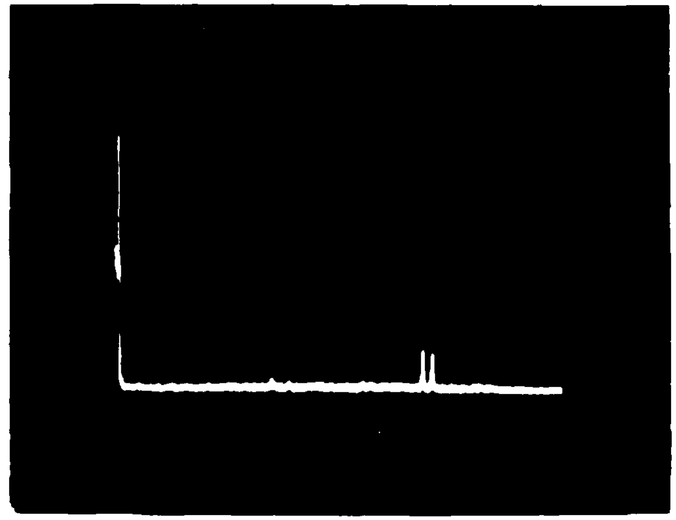
CURRENT SPECTRUM 200Hz/DIV REL AMP =1/DIV

FIGURE 4.3



A. 10 Hz, NL

CURRENT SPECTRUM 50Hz/DIV
REL AMP = 1/DIV

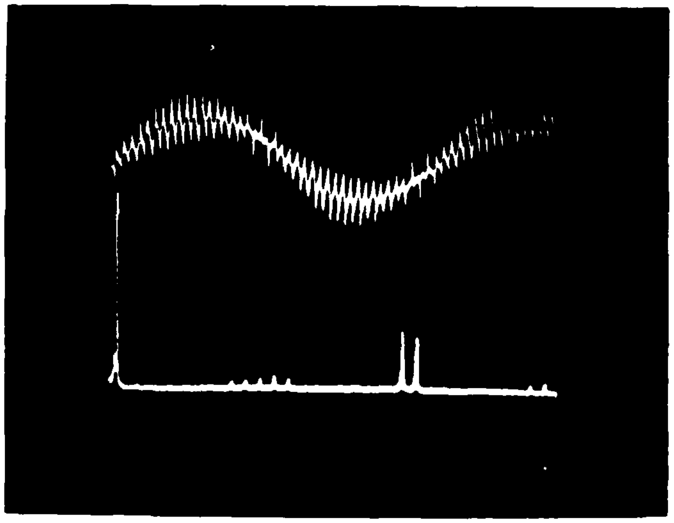


C. 20Hz, NL

CURRENT SPECTRUM 200Hz/CM
REL AMP = 1/DIV

FIGURE 4.4

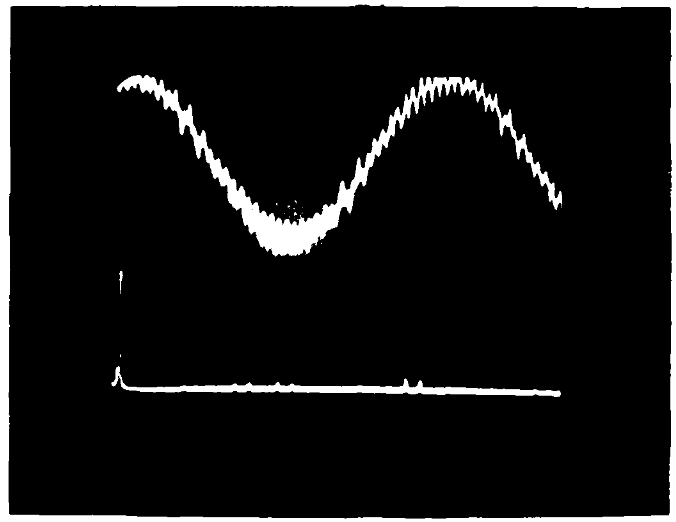
B. 30Hz, NL



OUTPUT CURRENT .5A/DIV

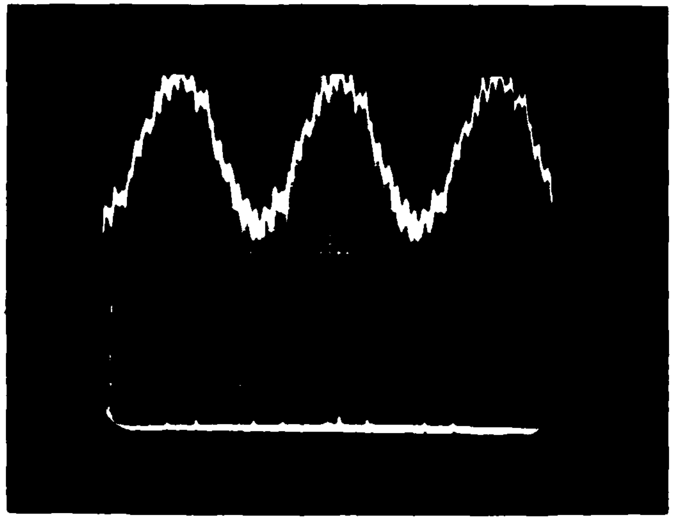
CURRENT SPECTRUM 200Hz/DIV
REL AMP = 1/DIV

D. 30Hz, 1/5 HP LOAD



OUTPUT CURRENT 1A/DIV

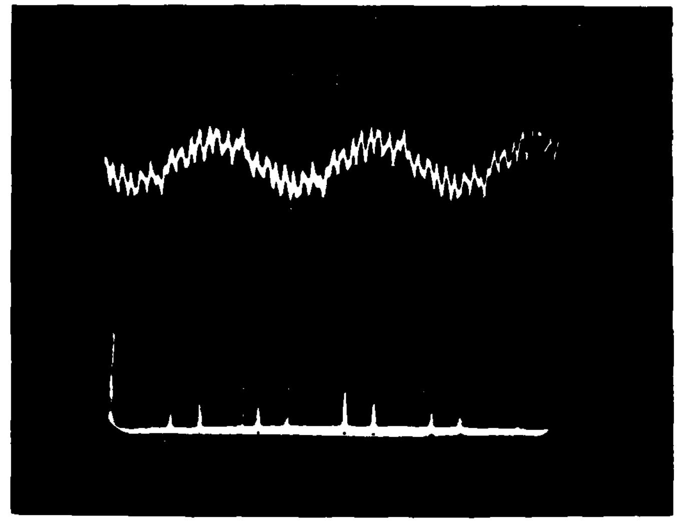
CURRENT SPECTRUM 200Hz/DIV
REL AMP = 3.3 /DIV



OUTPUT CURRENT 1A/DIV

CURRENT SPECTRUM 200Hz/DIV REL AMP =3.3/DIV

A. 60Hz, 1/5 HP LOAD

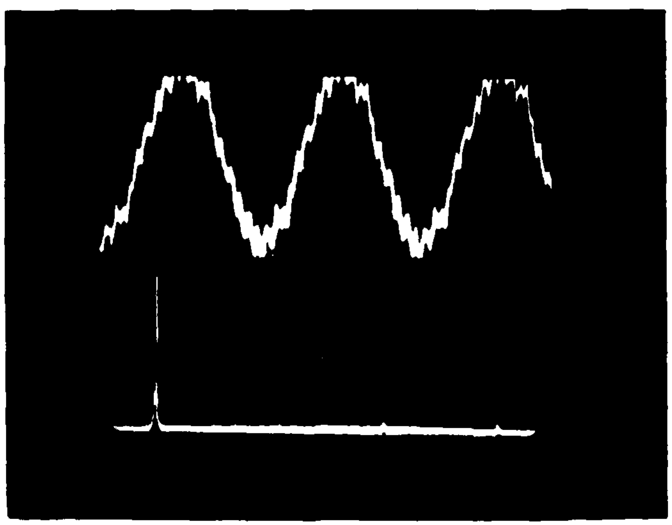


OUTPUT CURRENT 1A/DIV

CURRENT SPECTRUM 200Hz/DIV REL AMP = 1/DIV

C. 60Hz, NL

B. 60Hz, 1/5 HP LOAD



OUTPUT CURRENT 1A/DIV

CURRENT SPECTRUM 50Hz/DIV REL AMP =3.3/DIV

FIGURE 4.5

5. Conclusions & Recommendations

The Phase II development of the control circuits implement the theoretical requirements of the modulation and feedback mechanisms and provide a practical control circuits for a three phase motor drive. The feedback control produces efficient off-load operation as evidenced by the wide voltage swing between no load and full load on the small test motor. The feedback processor would appear to offer a means of sensing and control of both variable frequency, variable voltage AC drives and fixed frequency variable voltage drives (e.g. phase control of fan motors).

Performance to date has been very satisfactory and development of a full power system is recommended. Use of newer 16 bit microprocessors could permit extension of the control circuits to 400 Hz operation. Development of 400 Hz control and power stages is also recommended.

```
LOC  OBJ   SOURCE

                   NAM    PROTO BOARD MONITOR
           * PMON4


           0001    LSYMB  1
     6000 ACIAC    EQU    $6000          ;ACIA CONTROL REGISTER
     6001 ACIAD    EQU    $6001          ;ACIA DATA REG
     6000 ACIAS    EQU    $6000          ;ACIA STATUS REG
     0020 BLANK    EQU    $20            ;BLANK CHAR
     000D CR       EQU    $0D            ;CARRIAGE RET CHAR
     001B ESC      EQU    $1B            ;ABORT CHAR
     0004 EOT      EQU    $04            ;END OF MSG TO BE PRINTED
     FFFF LAST     EQU    $FFFF          ;HIGHEST ROM ADDRESS
     000A LF       EQU    $0A            ;LINE FEED
     007F RUBOUT   EQU    $7F            ;RUBOUT CHAR
          *
          *GLOBAL REF
          *  THESE LOCALS ARE NOT PROTECTED IF USER
          *   WRITES TO THEM IN USING MONITOR
          *
     0051 MONFG    EQU    $51            ;MONITOR CONNECTED FLAG(=1)
     00E6 NORFG    EQU    $E6            ;NORMAL OP FLAG =1MON,CONNECT
     00E7 VSFLG    EQU    $E7            ;VSTAR TEST FLAG=1(USE MONITOR)
          *  EXTERNALS FOR RSRSR AND PROM BURNER (NOT IMPLE.)
          *
          *       DEF MONENT,GETRNG,NXTADR,PXISTS,RNGERR,PBADR
          *       DEF PCRLF,OUTCH,PSPACE,SETMEM,ABORT
          *       DEF PROMAD,ADR,ADDL,ADDH,COUNT,MONITR
          *       DEF RSRSR,BURN,MOVE,READ,VFY,PINIT
          *
          *  RSRSR ROUTINE DEFINITIONS
          *
          *
          *
          *
          *  MONITOR RAM
          *
     FF90          ORG    $FFFE-110        ;*** CHANGE IF RAM USAGE CHAN
     FF90 BASE     EQU    *              ;BASE ADDR USED WITH INDEX OPS
     FF8F BOS      EQU    *-1            ;BOTTOM OF MONITOR STACK
          *
FF90      BUF      RMB    72             ;LINE OF TTY INPUT
          *
     FFD8 PROMAD   EQU    *              ;ADDRESS IN PROM
FFD8      OFFSET   RMB    2              ;OFFSET FOR LOADER/PUNCH
FFDA      ADR      RMB    2              ;PARAM. ENTERED BY USER
FFDC      ADDL     RMB    2
FFDE      ADDH     RMB    2
FFE0      BUFPTR   RMB    2              ;POINTER TO LAST CHAR SCANNED
FFE2      RECTYP   RMB    1              ;TAPE RECORD TYPE
FFE3      COUNT    RMB    1              ;COUNT FIELD FROM TAPE
FFE4      CKSM     RMB    1              ;CALCULATED CKSM
FFE5      SAVESP   RMB    2              ;TEMP STORAGE FOR S REG
FFE7      SAVEX    RMB    2              ;TEMP STORAGE FOR X REG
FFE9      ECHO     RMB    1              ;1=ECHO TTY, 0=NO ECHO
```

```
SEQ  LOC  OBJ       SOURCE

0056 FFEA           TCOUNT   RMB      1           ;TEMP LOC FOR COUNT
0057                *  USER REGISTERS
0058 FFEB           CREG     RMB      1
0059 FFEC           BREG     RMB      1
0060 FFED           AREG     RMB      1
0061 FFEE           XREG     RMB      2
0062 FFF0           PREG     RMB      2
0063 FFF2           SREG     RMB      2
0064                *
0065 FFF4           USWI     RMB      2           ;USER SWI VECTOR (NOT IMPLE.)
0066 FFF6           ACIAI    RMB      2           ;INDIRECT POINTER TO ACIA FOR RS..SR
0067 FFF8           IRQVEC   RMB      2           ;INTERRUPT REQUEST VECTOR
0068 FFFA           SWIVEC   RMB      2           ;SOFTWARE INTERRUPT VECTOR
0069 FFFC           NMIVEC   RMB      2           ;NON-MASKABLE INTERRUPT VECTOR
0070                *
0071                *
0072                ***    MONITOR ENTRY VECTOR    ***
0073                *
0074                *  RESTART INTERRUPT HANDLER
0075                *  INTERRUPT BREAK HANDLER
0076                *
0077                ***
0078           F000          ORG      $F000
0079                * RESET VECTOR ENTRY POINT
0080                *  JUMP IMMEDIATELY TO INITIAL SEQUENCE
0081 F000 7E F51C            JMP      PWRI        ;JUMP TO PWR ON INIT SEQ
0082                *
0083           F018          ORG      $F018
0084           F018 START    EQU      *           ;RESET INTERRUPT HANDLER
0085 F018 20 05             BRA      START1
0086           F01A BREAK    EQU      *           ;BREAK ON INTERRUPT ROUTINE
0087 F01A 7E F0CF            JMP      BREAK1
0088 F01D 60 00     ACIAA    FDB      ACIAC       ;POINTER TO ACIA
0089                *
0090           F01F START1   EQU      *
0091 F01F 36                 PSH A                ;SAVE AREG IF STACK EXISTS
0092 F020 07                 TPA                  ;SAVE CONDITION CODES
0093 F021 B7 FFEB            STA A    CREG
0094 F024 32                 PUL A
0095 F025 B7 FFED            STA A    AREG        ;SAVE CURRENT VALUE OF REGS
0096 F028 F7 FFEC            STA B    BREG
0097 F02B FF FFEE            STX      XREG        ;SAVE X
0098 F02E BF FFF2            STS      SREG        ;SAVE SP
0099 F031 8E FF8F            LDS      #BOS        ;INIT. SREG TO MON. STACK
0100 F034 CE F01A            LDX      #BREAK      ;BREAKPOINT ROUTINE
0101 F037 FF FFFC            STX      NMIVEC      ;STORE IN INTERRUPT VECTORS
0102 F03A FF FFF8            STX      IRQVEC
0103 F03D CE F029            LDX      #SWI30
0104 F040 FF FFF4            STX      USWI
0105 F043 CE F0D6            LDX      #SWIHAN     ;SOFTWARE INTERRUPT HANDLER
0106 F046 FF FFFA            STX      SWIVEC
0107 F049 CE F01D            LDX      #ACIAA      ;SET UP ACIA PTR
0108 F04C FF FFF6            STX      ACIAI
0109 F04F 86 03              LDA A    #3          ;RESET ACIA
0110 F051 B7 6000            STA A    ACIAC
```

```
SEQ  LOC  OBJ      SOURCE

0111                *
0112                * ACIA SET-UP :CLK/16,7 BITS+ODD PARITY+1 STOP BIT
0113                *
0114 F054 86 0D            LDA A   #$0D          ;SET ACIA CNTRL REG
0115 F056 B7 6000          STA A   ACIAC
0116                *  PRINT CR,LF, & RETURN TO MONITOR
0117                *
0118           F059 MONENT  EQU     *
0119           F059 MONEN1  EQU     *
0120 F059 BD F434          JSR     PINIT
0121 F05C BD F322          JSR     PCRLF
0122                *
0123                *     MONITOR ENTRY POINT
0124                *
0125                *
0126           F05F MONITR  EQU     *
0127 F05F 8E FF8F          LDS     #BOS          ;INIT MON.STACK
0128 F062 BD F3B9          JSR     RDROFF
0129 F065 B6 6001          LDA A   ACIAD         ;DUMP TTY INPUT DATA
0130 F068 86 3E            LDA A   #'>           ;PROMPT USER
0131 F06A BD F2FB          JSR     OUTCH
0132                *
0133                *  READ TTY LINE (BUFPTR)
0134                *   STORE TTY INPUT IN BUF UNTIL CR IS HIT
0135                *
0136 F06D CE FF90          LDX     #BUF          ;INITIALIZE BUFPTR
0137 F070 FF FFE0          STX     BUFPTR
0138 F073 0D               SEC                   ;SET ECHO FLAG
0139 F074 79 FFE9          ROL     ECHO
0140                *  BEGIN UNTIL LOOP
0141 F077 8C FFD7 RT10     CPX     #BUF+71       ;TEST FOR BUF OVERFLOW
0142 F07A 26 02            BNE     RT20          ;NO OVERFLOW
0143 F07C 20 47            BRA     ABORT
0144 F07E BD F41E RT20     JSR     WAITTY        ;READ NEXT CHAR
0145 F081 A7 00   RT30     STA A   0,X           ;INSERT CHAR INTO BUF
0146 F083 08               INX
0147                *  WHILE CONDITION
0148 F084 81 0D   RT90     CMP A   #CR           ;CARRIAGE RETURN?
0149 F086 26 EF            BNE     RT10          ;NO,CONTINUE LOOP
0150                *  END OF LOOP
0151                *
0152                *  DECODE ONE CHAR COMMAND
0153                *   COMPARE CHAR WITH TABLE OF VALID CHARS FOLLOWED BY
0154                *   ADDRESSES OF APPROPRIATE ROUTINES
0155                *
0156 F088 BD F3A3          JSR     PXISTS        ;GET 1ST CHAR
0157 F08B 08               INX                   ;INC BUFPTR
0158 F08C FF FFE0          STX     BUFPTR
0159 F08F CE F0A4          LDX     #CTABLE       ;START OF TABLE
0160                *  BEGIN LOOP
0161 F092 A1 00   DLOOP    CMP A   0,X           ;COMPARE
0162 F094 26 04            BNE     DL10
0163                *  FOUND CHAR. GET ADDRESS IMMEDIATELY FOLLOWING CHAR.
0164 F096 EE 01            LDX     1,X
0165 F098 6E 00            JMP     0,X           ;GO TO PROPER ROUTINE
```

```
LOC  OBJ       SOURCE

                 *  NO COMPARE.  MOVE TO NEXT CHAR.
F09A 08          DL10    INX
F09B 08                  INX
F09C 08                  INX
F09D 8C F0C5             CPX     #CTEND        ;END OF TABLE?
F0A0 26 F0               BNE     DLOOP         ;NO, REPEAT
                 *  END LOOP
F0A2 20 21               BRA     ABORT         ;NOT IN TABLE
                 *
          F05F   MONEND  EQU     MONITR
                 *
                 *  CTABLE: TABLE OF VALID 1 CHARACTER COMMANDS.
                 *     EACH ENTRY CONSISTS OF 3 BYTES.  BYTE 1
                 *     CONTAINS THE ASCII CHAR.  BYTES 2,3 CONTAIN THE
                 *     ADDRESS OF THE APPROPRIATE ROUTINE
                 *
          F0A4   CTABLE  EQU     *
F0A4 4C                  FCB     'L
F0A5 F1 CB               FDB     LOAD
F0A7 47                  FCB     'G
F0A8 F1 B7               FDB     GO
F0AA 50                  FCB     'P
F0AB F3 26               FDB     PUNCH
F0AD 42                  FCB     'B
F0AE F1 06               FDB     PREGS
F0B0 4D                  FCB     'M
F0B1 F5 A2               FDB     MOVE
F0B3 56                  FCB     'V
F0B4 F4 D3               FDB     VFY
F0B6 49                  FCB     'I
F0B7 F4 E0               FDB     READ
F0B9 53                  FCB     'S
F0BA F4 01               FDB     SM
F0BC 44                  FCB     'D
F0BD F1 4E               FDB     DM
F0BF 52                  FCB     'R
F0C0 F1 06               FDB     PREGS
F0C2 45                  FCB     'E
F0C3 F1 75               FDB     EOF
          F0C5   CTEND   EQU     *
                 *
                 *  ABORT
                 *
          F0C5   ABORT   EQU     *
          F0C5   BADINP  EQU     *
F0C5 CE F291             LDX     #MQUES        ;PRINT ????
                 *
                 *  PRINT MSG AND RETURN TO MONITOR
                 *
          F0C8   MSGMON  EQU     *
          F0C8   MSGABT  EQU     *
F0C8 8E FF8F             LDS     #BOS           ;S = BOTTOM OF STACK
F0CB 3F                  SWI
F0CC 12                  FCB     18            ;PMSG
F0CD 20 8A               BRA     MONEN1
```

```
SEQ  LOC  OBJ       SOURCE

                    *
                    *  SWI HANDLER :
                    *    DETERMINE  WHETHER SWI IS MONITOR CALL,RSRSR CALL,
                    *    OR USER SWI (NOTIMPLE.)
                    *
               F0CF BREAK1  EQU    *          ;BREAKPOINT ENTRY
     F0CF 01                NOP               ;REPLACES JSR PINIT
     F0D0 01                NOP               ;REPLACES JSR PINIT
     F0D1 01                NOP               ;REPLACES JSR PINIT
     F0D2 86 80             LDA A  #128       ;PRETEND TO BE SWI 128
     F0D4 20 1A             BRA    SWI40      ;SAVE REGS
                    *
               F0D6 SWIHAN  EQU    *
                    *  FIND INDEX BYTE (BYTE AFTER SWI THAT GOT US HERE)
     F0D6 30                TSX
     F0D7 EE 05             LDX    5,X        ;X = RET ADDR
     F0D9 A6 00             LDA A  0,X        ;A = INDEX BYTE
     F0DB 2B 0C             BMI    SWI30      ;BREAKPOINT?
                    *  IF USER HAS ADDITIONAL (RS)**3 ADDR OF 1ST+2 MUST BE DFF4
     F0DD 80 18             SUB A  #24        ;RSRSR CALL?
     F0DF 2A 03             BPL    SWI20      ;NO --
     F0E1 7E F5BE           JMP    RSRSR
                    *
                    *  USER SWI
                    *
     F0E4 FE FFF4 SWI20     LDX    USWI
     F0E7 6E 00             JMP    0,X        ;GO DO IT
                    *
                    *  MONITOR CALL. COPY REGS FROM STACK
                    *
     F0E9 30        SWI30   TSX               ;INCREMENT RET ADDR
     F0EA 6C 06             INC    6,X
     F0EC 26 02             BNE    SWI40
     F0EE 6C 05             INC    5,X
     F0F0 CE FFEB SWI40     LDX    #CREG      ;DEST. FOR 1ST REG
                    *  BEGIN LOOP
     F0F3 33        SWI50   PUL B             ;GET REG
     F0F4 E7 00             STA B  0,X        ;COPY
     F0F6 08                INX               ;MOVE TO NEXT REG
     F0F7 8C FFF2           CPX    #CREG+7    ;END OF LOOP?
     F0FA 26 F7             BNE    SWI50
                    *  END LOOP
                    *
                    *  S NOW CONTAINS ITS VALUE BEFORE SWI
                    *  WAS EXECUTED. SAVE IT
                    *
     F0FC AF 00             STS    0,X
                    *
                    *  A STILL CONTAINS SWI INDEX. TEST IT
                    *
     F0FE 81 81             CMP A  #129
     F100 26 04             BNE    PREGS      ;NOT 129 : BREAK
     F102 8D 07             BSR    PR1        ;129 : SNAPSHOT
     F104 20 1E             BRA    RESTAK     ;AND RETURN TO USER PROGRAM
                    *
```

```
SEQ  LOC  OBJ      SOURCE

                   *  PREGS : PRINT USER REGISTERS
                   *
          F106 PREGS    EQU    *
     F106 8D 03         BSR    PR1
     F108 7E F05F       JMP    MONEND
          F10B PR1      EQU    *          ;SUBR TO PRINT REGISTERS
     F10B CE FFEB       LDX    #CREG      ;X POINTS TO 1ST BYTE OF AREA
                   *  PRINT 3 1-BYTE REGS
     F10E C6 03         LDA B  #3         ;SET UP COUNT
                   *
     F110 3F       PR10     SWI
     F111 0F            FCB    15         ;P2HEX
     F112 BD F39E       JSR    PSPACE
     F115 5A            DEC B
     F116 2E F8         BGT    PR10
                   *
                   *  PRINT 3 2-BYTE REGS
     F118 C6 03         LDA B  #3         ;SET UP COUNT
                   *
     F11A BD F39A PR20  JSR    P4HEXS
     F11D 5A            DEC B
     F11E 2E FA         BGT    PR20
                   *
     F120 BD F322       JSR    PCRLF      ;PRINT CR & LF
     F123 39            RTS
                   *
                   *
                   *  RESTORE USER STATUS AND RETURN FROM MONITOR
                   *
                   *
     F124 BE FFF2 RESTAK LDS   SREG       ;TOP OF USER STACK
     F127 CE FFF1       LDX    #CREG+6    ;USER REGS.
                   *  BEGIN LOOP
     F12A A6 00    RUS10    LDA A  0,X    ;GET USER REG
     F12C 36            PSH A             ;PUSH INTO USER STACK
     F12D 09            DEX               ;MOVE TO NEXT REG
     F12E 8C FFEA       CPX    #CREG-1    ;LAST REG ?
     F131 26 F7         BNE    RUS10      ;NO, CONTINUE LOOP
                   *  END OF LOOP
     F133 3B            RTI               ;RETURN TO USER PROG
                   *
                   *  COMMANDS AND SUBROUTINES
                   *
                   *
                   *  CHECKSUM (CKSM)
                   *   VALIDATE CKSM
                   *
          F134 CHEKSM   EQU    *
     F134 B6 FFE4       LDA A  CKSM       ;SAVE CALC. CKSM
     F137 36            PSH A
     F138 BD F2BC       JSR    NEXT2D     ;A . = NEXT BYTE FROM TAPE
     F13B 33            PUL B
     F13C 53            COM B             ;B. = CALC. CKSM
     F13D 11            CBA               ;B = TAPE CKSM?
     F13E 26 01         BNE    CS1        ;NO.
```

```
LOC  OBJ      SOURCE

F140 39                RTS
                *
F141 30         CS1    TSX                  ;X: = ADDR OF CALC. CKSM
F142 09                DEX
F143 3F                SWI
F144 0F                FCB     15           ;P2HEX
F145 BD F39E           JSR     PSPACE
F148 CE F296           LDX     #MCSER       ;PRINT "CKSM ERR"
F14B 7E F0C8           JMP     MSGABT
                *
                *  DM ADDL,ADDH  COMMAND
                *
          F14E  DM     EQU     *
F14E 8D 35             BSR     GETRNG       ;GET ADR RANGE FROM BUF
                *                           RETURNS ADDL,ADDH+1
                *  BEGIN OUTER LOOP
F150 CE FFDC    DM10   LDX     #ADDL
F153 BD F39A           JSR     P4HEXS       ;PRINT MEM(X),SPACE,INC X
                *  BEGIN INNER LOOP
F156 FE FFDC    DM20   LDX     ADDL
F159 3F                SWI
F15A 0F                FCB     15           ;P2HEX
F15B BD F39E           JSR     PSPACE
F15E FF FFDC           STX     ADDL
F161 BC FFDE           CPX     ADDH         ;ADDL=ADDH+1,END RNGE
F164 27 0C             BEQ     DM50         ;EXIT OUTER LOOP
F166 B6 FFDD           LDA A   ADDL+1       ;LSB'S OF ADDL =0,END LINE
F169 84 0F             AND A   #$F
F16B 26 E9             BNE     DM20         ;NOT END OF LINE CONTINUE
                *  END OF INNER LOOP
F16D BD F322           JSR     PCRLF        ;PRINT CR,LF
F170 20 DE             BRA     DM10         ;EXIT INNER LOOP
                *  END OF OUTER LOOP
F172 7E F059    DM50   JMP     MONEN1       ;CR,LF, BACK TO MONITOR
                *
                *  PUNCH END OF FILE ANF 60 NULLS
                *
F175 CE F2A8    EOF    LDX     #MPEOF       ;PUNCH EOF RECORD
F178 3F                SWI
F179 12                FCB     18           ;PMSG
                *
                *  PUNCH 60 NULLS
                *
F17A C6 3B      NULLS  LDA B   #59          ;LOAD COUNT
                *  BEGIN LOOP
                *
F17C 4F         NULL1  CLRA                 ;LOAD NULL
F17D BD F2FB           JSR     OUTCH        ;PRINT 1 NULL
F180 5A                DEC B                ;DECR LOOP CNT
F181 26 F9             BNE     NULL1        ;DONE ?
                *  END OF LOOP
F183 20 ED             BRA     DM50         ;CR,LF, BACK TO MONITOR
                *
                *
                *  GETRANGE (ADDL,ADDH,BUFPTR)
```

```
LOC  OBJ       SOURCE

                    *    GET ADDRESS RANGE FROM BUF
                    *    ABORT IF INVALID
                    *    SET ADDH:= ADDH+1 TO SIMPLIFY COMPARISONS
                    *  RETURNS ADDL & ADDH+1
                    *  ALTERS ADDR,X,A,B
                    *
                    *
          F185 GETRNG  EQU    *
F185 BD F2D6           JSR    NXTADR        ;GET ADDR
F188 FE FFDA           LDX    ADR
F18B FF FFDC           STX    ADDL          ;STORE ADDL
F18E FF FFDE           STX    ADDH          ;MAY BE ONLY 1 PARAMETER
F191 BD F2D6           JSR    NXTADR        ;GET ADDH
F194 27 06             BEQ    GETRG3        ;ONLY 1 PARAMETER
                    *
F196 FE FFDA GETRG1    LDX    ADR
F199 FF FFDE           STX    ADDH          ;SAVE ADDH
                    *  THE NEXT 5 INSTR TEST ADDH-ADDL
F19C CE FF90 GETRG3    LDX    #BASE         ;REF W.R.T. BASE OF RAM
F19F A6 4E             LDA A  ADDH-BASE,X   ;MSBYTE
F1A1 E6 4F             LDA B  ADDH+1-BASE,X
F1A3 E0 4D             SUB B  ADDL+1-BASE,X
F1A5 A2 4C             SBC A  ADDL-BASE,X
F1A7 24 06             BCC    GETRG4        ;ADDH.GE.ADDL
F1A9 CE F283 RNGERR    LDX    #MRNGER       ;RANGE ERR  MSG
F1AC 7E F0C8           JMP    MSGABT        ;PRINT MSG & ABORT
                    *
F1AF FE FFDE GETRG4    LDX    ADDH          ;INC ADDH
F1B2 08                INX
F1B3 FF FFDE           STX    ADDH
F1B6 39                RTS
                    *
                    *   GO COMMAND
                    *
F1B7 CE F900 GO        LDX    #$F900        ;SET USER STACK VALUE
F1BA FF FFF2           STX    SREG          ;STORE IN USER ADDR
F1BD BD F2D6           JSR    NXTADR        ;GET PARAMETER
F1C0 27 06             BEQ    G10           ;NO PARAM,CONT EXCUT.
                    *
F1C2 FE FFDA           LDX    ADR           ;ADR=PARAM FROM NXTADR
F1C5 FF FFF0           STX    PREG
                    *
F1C8 7E F124 G10       JMP    RESTAK        ;(IN INTERRUPT HANDLER)
                    *
                    *  LOAD COMMAND
                    *
          F1CB LOAD    EQU    *
F1CB CE 0000           LDX    #0            ;INITIALIZE RANGE & OFFSET
F1CE FF FFD8           STX    OFFSET        ;TO 0000-FFFF,0000
F1D1 FF FFDC           STX    ADDL
F1D4 09      LOOFST    DEX
F1D5 FF FFDE           STX    ADDH
F1D8 BD F2D6           JSR    NXTADR        ;ANY OPERANDS?
F1DB 27 1E             BEQ    LHF2          ;NO,USE DEFAULT
F1DD FE FFDA           LDX    ADR           ;YES
```

```
SEQ  LOC  OBJ       SOURCE

0441 F1E0 FF FFD8            STX    OFFSET        ;IF ONE,IT'S OFFSET
0442 F1E3 BD F2D6            JSR    NXTADR        ;ANOTHER?
0443 F1E6 27 13              BEQ    LHF2          ;NO
0444 F1E8 FE FFD8            LDX    OFFSET        ;YES,1ST TWO ARE RANGE
0445 F1EB FF FFDC            STX    ADDL
0446 F1EE CE 0000            LDX    #0
0447 F1F1 FF FFD8            STX    OFFSET
0448 F1F4 8D A0              BSR    GETRG1
0449 F1F6 FE FFDE            LDX    ADDH
0450 F1F9 20 D9              BRA    LOOFST        ;GO TRY AGAIN FOR OFFSET
0451                *  BEGIN OUTER LOOP
0452 F1FB BD F3C3 LHF2       JSR    RDRON         ;TURN ON READER
0453                *  SHORT LOOP TO SKIP HDR RECORDS
0454 F1FE 8D 70    RDPRE     BSR    FINDS         ;FIND START OF RECORD
0455                *                   SETS (ECHO):=0 ON ENTRY
0456 F200 BD F41E            JSR    WAITTY        ;RETURNS (A):=TTY I/P
0457 F203 81 30              CMP A  #'0           ;IGNORE HDR RECORDS
0458 F205 27 F7              BEQ    RDPRE
0459                *  END SHORT LOOP
0460 F207 B7 FFE2            STA A  RECTYP        ;SAVE RECORD TYPE
0461 F20A 7F FFE4            CLR    CKSM
0462 F20D BD F2BC            JSR    NEXT2D        ;READ BYTE COUNT FROM TAPE
0463 F210 4A                 DEC A
0464 F211 4A                 DEC A
0465 F212 4A                 DEC A
0466 F213 B7 FFE3            STA A  COUNT         ;SAVE BYTE COUNT
0467 F216 BD F2BC            JSR    NEXT2D        ;READ ADR FIELD FROM TAPE
0468 F219 B7 FFDA            STA A  ADR           ;1ST BYTE
0469 F21C BD F2BC            JSR    NEXT2D
0470 F21F BB FFD9            ADD A  OFFSET+1
0471 F222 B7 FFDB            STA A  ADR+1         ;2ND BYTE
0472 F225 B6 FFDA            LDA A  ADR           ;CARRY TO FIRST BYTE
0473 F228 B9 FFD8            ADC A  OFFSET
0474 F22B B7 FFDA            STA A  ADR
0475 F22E B6 FFE2            LDA A  RECTYP        ;GET RECORD TYPE (0,1,9)
0476 F231 81 31    LHF3      CMP A  #'1           ;DATA RECORD?
0477 F233 26 14              BNE    LHF4          ;NO
0478                *
0479                *LOAD DATA RECORD
0480                *
0481                *  BEGIN UNTIL LOOP
0482                *
0483 F235 BD F2BC LDR10      JSR    NEXT2D        ;READ 2 HEX DIGITS FROM
0484                *                           TAPE. RETURNS IN A
0485 F238 FE FFDA            LDX    ADR
0486 F23B BD F3CD            JSR    SETOFF        ;STORE IN MEM(X),VERIFY
0487 F23E 08                 INX
0488 F23F FF FFDA            STX    ADR
0489 F242 7A FFE3            DEC    COUNT         ;DOES COUNT = 0
0490 F245 2E EE              BGT    LDR10         ;NO,CONT LOOP
0491                *  END UNTIL LOOP
0492 F247 20 04              BRA    LHF9
0493 F249 81 39    LHF4      CMP A  #'9           ;EOF RECORD?
0494 F24B 26 13              BNE    BADTAP        ;ILLEGAL RECORD TYPE
0495                *
```

```
LOC  OBJ       SOURCE

F24D BD F134 LHF9     JSR    CHEKSM        ;CHECK CKSM
F250 B6 FFE2          LDA A  RECTYP        ;GET RECORD TYPE
F253 81 39            CMP A  #'9           ;EOF RECORD?
F255 26 A4            BNE    LHF2          ;NO,CONT LOOP
                 *
                 *  END OF OUTER LOOP
                 *
F257 BD F3B9          JSR    RDROFF
F25A CE F28D          LDX    #MEOF         ;PRINT "EOF"
F25D 7E F0C8          JMP    MSGMON        ;AND RETURN TO MONITOR LOOP
                 *
F260 BD F3B9 BADTAP   JSR    RDROFF
                 *
F263 CE F29F          LDX    #MTAPER       ;PRINT "TAPE ERR"
F266 3F               SWI
F267 12               FCB    18            ;PMSG
                 *
                 *  ACCEPT NO COMMANDS UNTIL USER PRESSES ESC
                 *
F268 7C FFE9          INC    ECHO          ;SET ECHO
                 *
F26B BD F41E BT1      JSR    WAITTY        ;ESC CAUSES ABORT
F26E 20 FB            BRA    BT1
                 *
                 *  FIND S
                 *   READ TAPE UNTIL START OF RECORD
                 *
        F270 FINDS    EQU    *
F270 7F FFE9          CLR    ECHO          ;NO ECHO
                 *  BEGIN LOOP
F273 BD F41E FS10     JSR    WAITTY        ;READ NEXT TAPE CHAR
F276 81 53            CMP A  #'S           ;CHAR = S ?
F278 26 F9            BNE    FS10          ;NO
                 *  END LOOP
F27A 39               RTS
                 *
                 *  MESSAGES
                 *
F27B 42 4144 MBADR    FCC    /BAD ADR/
F282 04               FCB    $04
F283 52 414E MRNGER   FCC    /RANGE ERR/
F28C 04               FCB    $04
F28D 45 4F46 MEOF     FCC    /EOF/
F290 04               FCB    $04
F291 3F 3F3F MQUES    FCC    /????/
F295 04               FCB    $04
F296 43 4B53 MCSER    FCC    /CKSM ERR/
F29E 04               FCB    $04
F29F 54 4150 MTAPER   FCC    /TAPE ERR/
F2A7 04               FCB    $04
F2A8 53 3930 MPEOF    FCC    /S9030000FC/
F2B2 04               FCB    $04
F2B3 0D      MCRLFS   FCB    CR
F2B4 0A               FCB    LF
F2B5 00               FCB    $00
```

```
LOC  OBJ          SOURCE

F2B6 00                      FCB     $00
F2B7 00                      FCB     $00
F2B8 00                      FCB     $00
F2B9 53                      FCB     'S
F2BA 31                      FCB     '1
F2BB 04                      FCB     $04
                        *
                        *  NEXT 2 DIGITS--
                        *    READ NEXT 2 CHAR FROM TTY TAPE AND
                        *    CONVERT TO HEX NUMBER IN A REG. UPDATE CKSM.
                        *    RETURN UPDATED CKSM IN B REG.
                        *
          F2BC NEXT2D   EQU      *
F2BC BD F41E                 JSR     WAITTY        ;GET CHAR
F2BF 16                      TAB                   ;SAVE CHAR IN A
F2C0 BD F41E                 JSR     WAITTY
                        *
                        *  SET UP PARAMS FOR CONVERSION ROUTINE,
                        *  PUSH ASCII CHARS INTO STACK,POINT X AT STACK
                        *  SET A=TYPE OF CONVERSION AND B=# OF CHARS TO CONVERT
                        *
F2C3 36                      PSH A
F2C4 37                      PSH B
F2C5 30                      TSX
F2C6 C6 02                   LDA B   #2
F2C8 3F                      SWI
F2C9 15                      FCB     21            ;CONHB CNVT ASCII TO BINARY
F2CA 24 94                   BCC     BADTAP        ;IF NON-HEX CHAR. ABORT
                        *
F2CC 17                      TBA                   ;UPDATE CHECKSUM
F2CD FB FFE4                 ADD B   CKSM
F2D0 F7 FFE4                 STA B   CKSM
F2D3 31                      INS                   ;RESTORE STACK POINTER
F2D4 31                      INS
F2D5 39                      RTS
                        *
                        *  NEXT ADR(BUFPTR,ADR)
                        *
                        *    SET ADR=0 OR NEXT NUMBER STRING STARTING
                        *    AT BUFPTR
                        *    LEAVES BUFPTR AT CR.DELIMITER,OR 1ST
                        *    CHAR BETWEEN G - Z .
                        *    LEAVES (A)= LAST CHAR SCANNED
                        *    LEAVES (B)= LS BYTE OF ADR
                        *
                        *    RETURNS: CC=Z FOR NO PARAMETER
                        *             ABORTS IF NON-HEX PARAMETER
                        *
          F2D6 NXTADR   EQU      *
F2D6 7F FFDA                 CLR     ADR           ;ADR=0
F2D9 7F FFDB                 CLR     ADR+1
F2DC BD F3A3                 JSR     PXISTS        ;IS THERE A PARAMETER?
F2DF 26 01                   BNE     NA1           ;YES
F2E1 39                      RTS                   ;RETURN W/NO PARAM CC=Z
                        *
```

```
LOC  OBJ       SOURCE

               *  SET UP PARAMS FOR ASCII TO HEX CONVERSION
               *
F2E2 C6 47     NA1     LDA B  #71          ;MAX. CHARS TO SCAN
F2E4 3F                SWI
F2E5 15                FCB    21           ;CONHB
F2E6 FF FFE0           STX    BUFPTR
F2E9 B7 FFDA           STA A  ADR          ;SAVE RESULT
F2EC F7 FFDB           STA B  ADR+1
F2EF A6 00             LDA A  0,X          ;CHECK TERMINATOR
F2F1 3F                SWI
F2F2 13                FCB    19           ;ALPNUM, IS CHAR ALPHA?
F2F3 25 01             BCS    NA3          ;YES
F2F5 39                RTS
               *
F2F6 7E F0C5   NA3     JMP    ABORT        ;NO
               *
               *  OUTCH = PRINT CHAR IN A
               *  OUTCHX = PRINT CHAR AT  MEM(X)
               *    IF CHAR = 'CR',FOLLOW WITH LF & 4 NULLS
               *
F2F9 A6 00     OUTCHX  LDA A  0,X          ;ENTRY1
               *
          F2FB OUTCH   EQU    *
               * FIRST CHECK FOR ESC
F2FB 37                PSH B
F2FC F6 6000           LDA B  ACIAS        ;ACIA INPUT STATUS
F2FF 57                ASR B               ;C: = RDRF
F300 24 0A             BCC    OC10         ;NO INPUT
F302 F6 6001           LDA B  ACIAD        ;READ ACIA
F305 C1 1B             CMP B  #ESC
F307 26 03             BNE    OC10         ;NOT ESC
F309 7E F0C5           JMP    ABORT
               *
F30C 3F        OC10    SWI                 ;PRINT CHAR
F30D 11                FCB    17           ;PUTA
F30E 81 0D             CMP A  #CR
F310 26 0E             BNE    OC20         ;NOT CR. RETURN
               *
F312 86 0A             LDA A  #LF          ;PRINT LF
F314 3F                SWI
F315 11                FCB    17           ;PUTA
F316 4F                CLR A               ;PRINT 4 NULLS
F317 C6 04             LDA B  #4           ;SET LOOP CNT
               *  BEGIN LOOP
F319 3F        OCLOOP  SWI
F31A 11                FCB    17           ;PUTA
F31B 5A                DEC B
F31C 26 FB             BNE    OCLOOP
               *  END LOOP
F31E 86 0D             LDA A  #CR          ;RESTORE A
               *
F320 33        OC20    PUL B
F321 39                RTS
               *
               *  PRINT CR,LF,NULL
```

```
LOC  OBJ       SOURCE

               *
               *
F322 86 0D     PCRLF   LDA A   #CR
F324 20 D5             BRA     OUTCH         ;OUTCH PRINTS LF AFTER CR
               *
               *  PUNCH  ADDL,ADDH
               *    PUNCH MEMORY CONTENTS BETWEEN ADDL & ADDH
               *    IN HEX FORMAT
               *
F326 BD F185   PUNCH   JSR     GETRNG        ;READ ADDL & ADDH+1
F329 CE 0000           LDX     #0
F32C FF FFD8           STX     OFFSET
F32F 8D A5             BSR     NXTADR        ;ANY OFFSET?
F331 27 06             BEQ     PHF15         ;NO
F333 FE FFDA           LDX     ADR           ;YES
F336 FF FFD8           STX     OFFSET
               *
               *  PUNCH DATA RECORD UNTIL ADDL = ADDH
               *
          F339 PHF15   EQU     *
               *  BEGIN LOOP
               *
               *  CALCULATE DATA LENGTH = MIN(30,ADDH+1-ADR)
               *
F339 F6 FFDF   PHF20   LDA B   ADDH+1        ;B:=ADDH-ADDL
F33C F0 FFDD           SUB B   ADDL+1
F33F B6 FFDE           LDA A   ADDH
F342 B2 FFDC           SBC A   ADDL
F345 26 04             BNE     PUND10        ;DIFF GT 256
F347 C1 1E             CMP B   #30           ;LS BYTE GT  30?
F349 23 02             BLS     PUND20
               *
F34B C6 1E     PUND10  LDA B   #30           ;DIFF GT  30
               *
F34D 5C        PUND20  INC B                 ;COUNT:=COUNT+3
F34E 5C                INC B                 ;...INCLUDES ADDR & CKSM
F34F 5C                INC B
F350 F7 FFE3           STA B   COUNT
F353 CE F2B3           LDX     #MCRLFS
F356 3F                SWI
F357 12                FCB     18            ;PMSG
F358 5F                CLR B                 ;B HOLDS CKSM
F359 CE FFE3           LDX     #COUNT
F35C 8D 34             BSR     PUNBYT
F35E 37                PSH B
F35F FE FFDC           LDX     ADDL          ;COMPUTE OFFSET ADDRESS
F362 B6 FFD8           LDA A   OFFSET
F365 F6 FFD9           LDA B   OFFSET+1
F368 3F                SWI
F369 08                FCB     8             ;ADDABX
F36A FF FFDA           STX     ADR           ;PUNCH FROM ADR
F36D CE FFDA           LDX     #ADR
F370 33                PUL B
               *
F371 8D 1F             BSR     PUNBYT        ;(INCREMENTS X)
```

```
SEQ  LOC  OBJ        SOURCE

     F373 8D 1D              BSR     PUNBYT
     F375 FE FFDC            LDX     ADDL          ;RESTORE X
                      *
                      *  PUNCH BYTES FROM MEMORY UNTIL COUNT IS EXHAUSTED
                      *
                      *  BEGIN LOOP
     F378 8D 18       PREC10 BSR     PUNBYT        ;(CC=0 IF COUNT = 0)
     F37A 2E FC              BGT     PREC10
                      *  END LOOP
     F37C FF FFDC            STX     ADDL          ;SAVE X
     F37F CE FFE4            LDX     #CKSM         ;PUNCH CHSM
     F382 53                 COM B
     F383 E7 00              STA B   0,X           ;CKSM:=B
     F385 8D 0B              BSR     PUNBYT
     F387 FE FFDC            LDX     ADDL
     F38A BC FFDE            CPX     ADDH
     F38D 26 AA              BNE     PHF20
                      *  END LOOP
     F38F 7E F059            JMP     MONEN1
                      *
                      *  PUNBYT (MEM(X),COUNT,CKSM)
                      *     PUNCH BYTE AT MEM(X) AND ADJUST COUNT AND CKSM
                      *     CC=Z IF COUNT =0
                      *
     F392 EB 00       PUNBYT ADD B   0,X           ;CKSM:=CKSM+MEM(X)
     F394 3F                 SWI
     F395 0F                 FCB     15            ;P2HEX,PRINT MEM(X) AS 2 CHAR
     F396 7A FFE3            DEC     COUNT
     F399 39                 RTS
                      *
                      *  P4HEXS:  PRINT 2 BYTES AT X AS 4 HEX CHARS + 2 SPACES
                      *
     F39A 3F          P4HEXS SWI
     F39B 10                 FCB     16            ;P4HEX
     F39C 8D 00              BSR     PSPACE
                      *
                      *  PSPACE--- PRINT 1 BLANK SPACE
                      *
     F39E 86 20       PSPACE LDA A   #BLANK
     F3A0 3F                 SWI
     F3A1 11                 FCB     17            ;PUTA
     F3A2 39                 RTS
                      *
                      *  PARAM EXISTS(BUFPTR)     (#BUFPTR)= BUFPTR
                      *                              (X) = BUFPTR
                      *  INC BUFPTR UNTIL CHAR = ALPHA OR CR
                      *  LEAVE A = MEM(BUFPTR)
                      *  SET Z IF NO PARAMETER EXISTS
                      *
          F3A3        PXISTS EQU     *             ;ENTRY FOR(#BUFPTR)=BUFPTR
     F3A3 FE FFE0            LDX     BUFPTR
          F3A6        PXISTX EQU     *             ;ENTRY FOR (X)=BUFPTR
                      *  BEGIN LOOP
     F3A6 A6 00       PX1    LDA A   0,X           ;IS CHAR ALPHANUM?
     F3A8 3F                 SWI
```

```
SEQ  LOC  OBJ       SOURCE

     F3A9 13                   FCB     19          ;ALPNUM
     F3AA 25 07                BCS     PX2         ;YES,EXIT LOOP
     F3AC 81 0D                CMP A   #CR         ;IS CHAR CR?
     F3AE 27 03                BEQ     PX2         ;YES,EXIT LOOP
     F3B0 08                   INX                 ;MOVE TO NEXT CHAR
     F3B1 20 F3                BRA     PX1
                    *  END LOOP
     F3B3 FF FFE0 PX2          STX     BUFPTR
     F3B6 81 0D                CMP A   #CR         ;SET Z IF NO PARAMETER
     F3B8 39                   RTS
                    *
                    *  RDR OFF
                    *    TURNS TAPE RDR OFF:
                    *          ACIA RTS O/P HIGH
                    *          ACIA CHAR $2A (*)
                    *  OLD VERSION OUTPUT ASCII 13 (DC3)
          F3B9 RDROFF  EQU     *
     F3B9 86 01                LDA A   #$01        ;RTS HIGH
     F3BB B7 6000 RDF90        STA A   ACIAC       ;SET ACIA CONT REG
     F3BE 86 2A                LDA A   #$2A        ;SEND TTY RDR CONT CHAR
     F3C0 3F                   SWI
     F3C1 11                   FCB     17          ;PUTA
     F3C2 39                   RTS
                    *
                    *  RDR ON:
                    *    TURNS ON TAPE READER
                    *       ACIA RTS O/P LOW
                    *       ACIA CHAR $11 (DC1)
                    *
          F3C3 RDRON   EQU     *
     F3C3 86 41                LDA A   #$41        ;RTS LOW
     F3C5 B7 6000 RON90        STA A   ACIAC       ;SET ACIA CONT REG
     F3C8 86 11                LDA A   #$11        ;SEND TTY RDR CONT CHAR
     F3CA 3F                   SWI
     F3CB 11                   FCB     17          ;PUTA
     F3CC 39                   RTS
                    *
                    *  SETMEM(X)
                    *  SETS MEM(X):=A AND VERIFY
                    *
          F3CD SETOFF  EQU     *
     F3CD 36                   PSH A               ;FIRST CHECK RANGE
     F3CE B6 FFDC              LDA A   ADDL        ;LOW LIMIT
     F3D1 F6 FFDD              LDA B   ADDL+1
     F3D4 3F                   SWI
     F3D5 06                   FCB     11          ;SUBXAB
     F3D6 22 0A                BHI     SETOUT      ;TOO LOW
     F3D8 B6 FFDE              LDA A   ADDH        ;HIGH LIMIT
     F3DB F6 FFDF              LDA B   ADDH+1
     F3DE 3F                   SWI
     F3DF 0B                   FCB     11          ;SUBXAB
     F3E0 24 07                BCC     SETFUL      ;OK
     F3E2 32           SETOUT  PUL A               ;OUTSIDE RANGE LIMITS
     F3E3 86 FF                LDA A   #255        ;TYPE DELETE(RUBOUT)
     F3E5 3F                   SWI
```

```
SEQ  LOC  OBJ       SOURCE

0826 F3E6 11                  FCB     17            ;PUTA     SIGNAL FACT TO USER
0827 F3E7 20 17               BRA     SETM1         ;OTHER,IGNORE STORE REQ
0828 F3E9 32        SETFUL    PUL A
0829           F3EA SETMEM    EQU     *
0830 F3EA A7 00               STA A   0,X
0831 F3EC A1 00               CMP A   0,X           ;VERIFY
0832 F3EE 27 10               BEQ     SETM1         ;ERROR?
0833                *  VERIFY ERROR,PRINT ADR
0834 F3F0 FF FFDA             STX     ADR           ;SET PARAM FOR P4HEX
0835 F3F3 CE FFDA             LDX     #ADR
0836 F3F6 8D C1               BSR     RDROFF
0837 F3F8 8D A0               BSR     P4HEXS
0838 F3FA CE F27B FBADR       LDX     #MBADR        ;PRINT "BAD ADR"
0839 F3FD 7E F0C8             JMP     MSGABT        ;PRINT MSG & ABORT
0840                *
0841 F400 39        SETM1     RTS
0842                *
0843                *  SM  ADR BYTE1,BYTE2,....
0844                *
0845           F401 SM        EQU     *
0846 F401 BD F206             JSR     NXTADR        ;ADR:= NEXT PARAM
0847                *
0848 F404 FE FFDA SM5         LDX     ADR           ;SAVE ADR IN ADL
0849 F407 FF FFDC             STX     ADDL
0850                *  BEGIN WHILE LOOP
0851 F40A BD F206 SM10        JSR     NXTADR        ;ADR:= NEXT PARAM
0852 F40D 27 0C               BEQ     SM30          ;END OF LINE,EXIT LOOP
0853 F40F FE FFDC             LDX     ADDL          ;X:= ADD TO BE SET
0854 F412 17                  TBA                   ;A:= LS BYTE
0855 F413 8D D5               BSR     SETMEM        ;MEM(X):=A,VERIFY
0856 F415 08                  INX                   ;MOVE TO NEXT ADD
0857 F416 FF FFDC             STX     ADDL
0858 F419 20 EF               BRA     SM10
0859                *  END OF LOOP
0860 F41B 7E F03F SM30        JMP     MONEND
0861                *
0862                *  WAIT FOR TTY(CHAR,ECHO)   (#ECHO)=ECHO
0863                *      RETURN NEXT TTY CHAR IN A
0864                *      IF (#ECHO) NOT 0 , ECHO CHAR
0865                *
0866           F41E WAITTY    EQU     *
0867                *  LOOP UNTIL INPUT .NE. RUBOUT
0868 F41E 3F        W10       SWI                   ;READ TTY
0869 F41F 14                  FCB     20            ;GETA
0870 F420 81 1B               CMP A   #ESC          ;ESCAPE?
0871 F422 26 03               BNE     W20           ;NO
0872 F424 7E F0C5             JMP     ABORT         ;YES,ABORT
0873 F427 81 7F     W20       CMP A   #RUBOUT       ;RUBOUT?
0874 F429 27 F3               BEQ     W10           ;YES CONTINUE
0875                *  END UNTIL LOOP
0876 F42B 7D FFE9             TST     ECHO
0877 F42E 27 03               BEQ     W30           ;NO ECHO
0878 F430 BD F2FB             JSR     OUTCH         ;ECHO A
0879 F433 39        W30       RTS
0880                * WAS  ORG $F42E
```

```
SEQ  LOC  OBJ       SOURCE

0881           0001 MOVER    EQU    1             ;0=MOVE ROUTINE EXCLUDED
0882           000A DELAY    EQU    10            ;POST PROG DELAY BEFORE VFY
0883                *  PIA LOCATIONS
0884           5000 PIA      EQU    $5000
0885           0001 V50      EQU    $5001-PIA
0886           0004 PROM     EQU    $5004-PIA
0887                *  STANDARD RAM BUFFER (DEFAULT)
0888           FC00 RAM      EQU    $FC00
0889                *
0890                *RSRSR CALL LOCATIONS
0891                *
0892                *  INITIALIZE PROM BURNER PIA'S
0893           F434 PINIT    EQU    *
0894 F434 39                 RTS                  ;SUB FOR ORG MONITOR
0895                * WAS ORG $F430
0896                *
0897                *  TYPE A IN BINARY,ENCLOSED BY SPACES
0898                *
0899           F435 P8BIN    EQU    *
0900 F435 37                 PSHB                 ;SAVE B
0901 F436 36                 PSHA
0902 F437 8D 0F              BSR    PSP           ;PRINT LEADING SPACE
0903 F439 32                 PUL A
0904 F43A C6 08              LDA B  #8            ;8 DIGIT COUNTER
0905 F43C 49        A8       ROL A
0906 F43D 36                 PSH A
0907 F43E 86 18              LDA A  #24           ;(=1/2 ASCII "0")
0908 F440 49                 ROL A
0909 F441 3F                 SWI
0910 F442 11                 FCB    17            ;PRINTA
0911 F443 32                 PUL A
0912 F444 5A                 DEC B
0913 F445 26 F5              BNE    A8
0914 F447 33                 PUL B
0915 F448 7E F39E PSP        JMP    PSPACE        ;PRINT ONE MORE SPACE
0916                *
0917                *  RAM/ROM ADDRESS SETUP & VALIDATION
0918                *
0919 F44B CE FC00 RASV       LDX    #RAM          ;INITIALIZE POINTERS TO DEFAULT RAM
0920 F44E FF FFDC            STX    ADDL
0921 F451 CE FE00            LDX    #RAM+512
0922 F454 FF FFDE            STX    ADDH
0923 F457 7F FFE3            CLR    COUNT         ;SET FULL PROM FLAG
0924 F45A BD F3A3            JSR    PXISTS        ;..IF NO ADDRESS
0925 F45D 27 06              BEQ    AA1
0926 F45F BD F185            JSR    GETRNG
0927 F462 7C FFE3            INC    COUNT
0928 F465 FE FFDC AA1        LDX    ADDL          ;DEFAULT PROM ADDRESS
0929 F468 FF FFD8            STX    PROMAD        ;IS SAME AS START
0930 F46B BD F2D6            JSR    NXTADR        ;TRY FOR PROM ADDRESS
0931 F46E 27 06              BEQ    AA3           ;NO
0932 F470 FE FFDA            LDX    ADR           ;YES
0933 F473 FF FFD8            STX    PROMAD
0934 F476 CE FFD8 AA3        LDX    #PROMAD       ;VERIFY THAT RANGE <= 512
0935 F479 0D                 SEC                  ;(FORCE BORROW)
```

```
SEQ  LOC  OBJ        SOURCE

     F47A E6 07              LDA B  7,X          ;=ADDH+1
     F47C E2 05              SBC B  5,X          ;=ADDL+1
     F47E A6 06              LDA A  6,X
     F480 A2 04              SBC A  4,X
     F482 81 02              CMP A  #2           ;SHOULD BE 1 OR 0
     F484 2C 08              BGE    AA4          ;TOO BIG
     F486 EB 01              ADD B  1,X          ;+ SHOULD NOT OVERSP PROM
     F488 A9 00              ADC A  0,X
     F48A A8 00              EOR A  0,X
     F48C 84 FE              AND A  #$FE
     F48E 26 01              BNE    AA4          ,IT DOES
     F490 39                 RTS
     F491 7E F1A3 AA4        JMP    RNGERR       ;ADDRESS RANGE ERROR
                        *
                        *  TYPE RAM & ROM ADDRESS & DATA
                        *
     F494 CE FFDC VERR       LDX    #ADDL        ;TYPE RAM ADDRESS
     F497 3F                 SWI
     F498 10                 FCB    16           ,P4HEX
     F499 FE FFDC            LDX    ADDL         ;NOW THE BYTE THERE
     F49C A6 00              LDA A  0,X
     F49E 8D 95              BSR    P8BIN
     F4A0 B6 5006            LDA A  PROM+2+PIA   ;THEN PROM DATA
     F4A3 8D 90              BSR    P8BIN
     F4A5 CE FFD8            LDX    #PROMAD      ;NOW IF ADDRESS (LOW 8)
     F4A8 A6 01              LDA A  1,X          ;DOES NOT MATCH RAM ADDRESS
     F4AA A1 05              CMP A  5,X          ;=ADDL
     F4AC 27 02              BEQ    AT
     F4AE 3F                 SWI                 ;PRINT PROM ADDRESS
     F4AF 10                 FCB    16           ;P4HEX
     F4B0 BD F322 AT         JSR    PCRLF
     F4B3 86 40              LDA A  #64
     F4B5 48                 ASL A               ;EXIT C=0,Z=0,V=1
     F4B6 39                 RTS
                        *
                        *  PROM ADDRESS SETUP & DATA READ
                        *
     F4B7 CE FFD8 ADDRS      LDX    #PROMAD
     F4BA A6 01              LDA A  1,X          ;LOW 8 BITS
     F4BC B7 5004            STA A  PROM+PIA
     F4BF A6 00              LDA A  0,X          ;HIGH BIT
     F4C1 CE 5000            LDX    #PIA
     F4C4 48                 ASL A               ;POSITION IT
     F4C5 4C                 INC A               ;WITH DATA REGISTER SELECT
     F4C6 48                 ASL A
     F4C7 48                 ASL A
     F4C8 A8 07              EOR A  PROM+3,X     ;INSERT INTO CONTROL
     F4CA 84 0C              AND A  #12
     F4CC A8 07              EOR A  PROM+3,X
     F4CE A7 07              STA A  PROM+3,X
     F4D0 A6 06              LDA A  PROM+2,X     ;READ DATA
     F4D2 39                 RTS
                        *
                        *  PROM VERIFY
                        *
```

```
LOC  OBJ       SOURCE

F4D3 BD F446 VFY      JSR    RASV         ;GO SET UP ADDRESSES
F4D6 8D 2D   AV       BSR    VFY1         ;VERIFY ONE LOCATION
F4D8 24 02            BCC    AN           ;NO ERROR,OR PRINTED
F4DA 8D 3A            BSR    JVER         ;PRINT FIXABLE ERROR.
F4DC 8D 11   AN       BSR    INCAD        ;INCREMENT ADDRESSES
F4DE 20 F6            BRA    AV
                      *
                      *  PROM READ
                      *
F4E0 BD F446 READ     JSR    RASV         ;SET UP POINTERS
F4E3 8D D2   AR       BSR    ADDRS        ;READ ONE BYTE
F4E5 FE FFDC          LDX    ADDL
F4E8 BD F1EA          JSR    SETMEM       ;STORE IN RAM
F4EB 8D 02            BSR    INCAD        ;NEXT
F4ED 20 F4            BRA    AR
                      *
                      *  INCREMENT RAM/ROM ADDRESS POINTERS
                      *
F4EF FE FFD8 INCAD    LDX    PROMAD
F4F2 08               INX
F4F3 FF FFD8          STX    PROMAD
F4F6 FE FFDC INK      LDX    ADDL
F4F9 08               INX
F4FA FF FFDC          STX    ADDL
F4FD BC FFDE          CPX    ADDH
F500 26 B5            BNE    ADDRS
F502 7E F03F EXIT     JMP    MONITR       ;EXIT TO MONITOR
                      *
                      *  PROM DATA VERIFY,ONE BYTE
                      *
F505 8D B0   VFY1     BSR    ADDRS        ;SET UP & READ A BYTE
F507 FE FFDC          LDX    ADDL         ;COMPARE TO RAM
F50A A1 00            CMP A  0,X
F50C 27 07            BEQ    AX           ;OK:C=0,Z=1,V=0
F50E 43               COM A               ;NO,IS IT FIXABLE
F50F AA 00            ORA A  0,X          ;I.E.  NO RAM=0,PROM=1?
F511 43               COM A
F512 26 02            BNE    JVER         ;YES.  C=1,Z=0,V=0
F514 4C               INC A
F515 39      AX       RTS
F516 7E F494 JVER     JMP    VERR         ;NO.  TYPE ERROR
                      *
                      *  PROM BURNER ROUTINE
                      *    NOT CODED
                      *WAS  ORG $F514
     F519    BURN     EQU    *
F519 7E F0C5          JMP    ABORT
                      *WAS ORG $F520
                      * POWER ON INITIAL SEQUENCE
                      *  SETS PIA TO DETERMINE MODE
                      *  PORT "A" = INPUT
                      *  PORT "B" = OUTPUT + 1 INPUT
                      *    PB0 - PB6 = OUTPUTS
                      *    PB7 = INPUT OF TTY ENABLE (0=TTY CONNECT)
     D800    OPMODE   EQU    $D800        ;NORMAL OP MODE FROM ADDR
```

```
SEQ  LOC  OBJ      SOURCE

               F51C FWRI     EQU     *
     F51C 86 00             LDA A   #$00         ;LOAD CRA WORD
     F51E B7 5001           STA A   $5001        ;SET CRA
     F521 B7 5000           STA A   $5000        ;ENB DDRA ALL INPUT
     F524 86 04             LDA A   #$04         ;SET PRA ENAB WORD
     F526 B7 5001           STA A   $5001        ;ENAB TO READ PRA
     F529 86 00             LDA A   #$00         ;SET CRB WORD
     F52B B7 5003           STA A   $5003        ;SET CRB
     F52E 86 7F             LDA A   #$7F         ;SET DDRB WORD
     F530 B7 5002           STA A   $5002        ;SET DDRB
     F533 86 04             LDA A   #$04         ;SET CRB WORD
     F535 B7 5003           STA A   $5003        ;ENAB PRB
     F538 86 60             LDA A   #$60         ;SET PRB WORD
     F53A B7 5002           STA A   $5002        ;SET PRB
     F53D B6 5000           LDA A   $5000        ;READ PORT A DATA
     F540 84 10             AND A   #$10         ;MASK ALL BUT TTY
     F542 27 0C             BEQ     MCINIT       ;BRANCH IF TTY CONNECTED
     F544 7F 0051           CLR     MONFG        ;RESET MONITOR CONN FLAG
     F547 7F 00E6           CLR     NORFG        ;RESET NORMAL FLAG
     F54A 7F 00E7           CLR     VSFLG        ;VSTAR TEST FLAG
     F54D 7E D800           JMP     OPMODE       ;JUMP IF NORMAL OP MODE
                    *
                    *  PTM2 SET-UP . SETS CR3 AND TIMER 3
                    *   TO GENERATE CLOCK FOR ACIA.
                    *   TIMER 3 CONTINUOUS,NO INTERRUPT (IRQ),
                    *   FREQUENCY = 1 MHZ/52/4 = 16*300 BAUD (4800 HZ.)
                    *   TIMER STARTS IMMEDIATELY(G-D+R+W)
     F550 86 81     MCINIT  LDA A   #$81         ;LOAD WORD FOR CR2
     F552 B7 3001           STA A   $3001        ;STORE IN CR2
     F555 86 80             LDA A   #$80         ;LOAD WORD FOR CR1,CR2
     F557 B7 3000           STA A   $3000        ;STORE IN CR1
     F55A B7 3001           STA A   $3001        ;STORE IN CR2
     F55D 86 82             LDA A   #$82         ;LOAD WORD FOR CR3
     F55F B7 3000           STA A   $3000        ;STORE IN CR3
     F562 CE 0068           LDX     #$0068       ;LD DIV CNT FOR T3
     F565 FF 3006           STX     $3006        ;STORE TO T3 DATA REG
     F568 86 01             LDA A   #01          ;SET MON CONN FLAG VALUE
     F56A 97 51             STA A   MONFG        ;STORE FLAG,MON. CONNECTED
     F56C 97 E6             STA A   NORFG        ;SET NORMAL OP WITH KEYBD
     F56E 97 E7             STA A   VSFLG        ;SET VSTAR TEST FLAG
     F570 7E F018           JMP     START        ;JUMP TO MON START
               F595         ORG     $F595
     F595 7E F3FA JBAD      JMP     PBADR        ;PRNT"BAD ADDRESS"& QUIT
               F5A2         ORG     $F5A2
                    *
                    *  MEMORY MOVE
                    *
     F5A2 BD F185 MOVE      JSR     GETRNG       ,GET SOURCE ADDRESS RANGE
     F5A5 BD F2D6           JSR     NXTADR       ,GET DEST STARTING ADD
     F5A8 27 EB             BEQ     JBAD         ,ERROR IF DONE
     F5AA FE FFDC AM        LDX     ADDL         ,GET BYTE
     F5AD A6 00             LDA A   0,X
     F5AF FE FFDA           LDX     ADR          ,STORE IT WITH VERIFY
     F5B2 BD F3EA           JSR     SETMEM
     F5B5 08                INX                  ,INCREMENT POINTERS
```

```
SEQ  LOC  OBJ       SOURCE

     F5B6 FF FFDA             STX     ADR
     F5B9 BD F4F6             JSR     INK          ;COMPARE TO END
     F5BC 20 EC               BRA     AM           ;MORE
                    *
                    *  END OF MODULE
                    *
          0018 NITEMS  EQU     24           ;NUMBER OF ROUTINES
                    *
                    *  CALLING SEQUENCE LOC
                    *                         X      SWI
                    *                         X+1    INDEX
                    *                         X+2    NEXT INSTRUCTION
          F5BE              ORG     $F5BE
                    *
                    *  ENTRY IS VIA LOW ORDER ADDRESS OF ROM
                    *    <<ADDRESS IS PLACED IN SWI VECTOR ADDRESS
          F5BE RSRSR   EQU     *
                    *
                    *  GET THE INDEX VALUE
                    *    DOUBLE IT FOR VECTOR ADDRESS INDEX
                    *
     F5BE 30                  TSX                  ;SP INTO X
                    *  RESTORE STATE OF INTERRUPT AT TIME OF CALL
                    *
     F5BF EE 05               LDX     5,X          ;X HAS INDEX ADDRESS
                    *
     F5C1 4F                  CLR A
     F5C2 E6 00               LDA B   0,X          ;INDEX INTO B
     F5C4 58                  ASL B                ;DOUBLE
     F5C5 49                  ROL A
                    *
                    *  A,B HAS TWO TIMES INDEX
                    *
                    *  VECTOR OF SUBROUTINE ADDRESSES IS AT 512+ROM BASE
                    *
                    *  FROM HERE TO VECTOR IS 512-WHERE WE ARE
     F5C6 8D 00               BSR     LOCVV
                    *
                    *
          F5C8 LOCVV   EQU     *                   ;STACK HAS WHERE WE ARE
     F5C8 30                  TSX
                    *  A,B WILL HAVE INDEX*2 + LOCATION(AA)
     F5C9 EB 01               ADD B   1,X
     F5CB A9 00               ADC A   0,X
                    *  ADD VECTOR OFFSET
                    *
     F5CD CB 24               ADD B   #$24            ;LOW ORDER 8 BITS
     F5CF 89 01               ADC A   #$01            ;HIGH ORDER 8 BITS
                    *
                    *  A,B NOW HAVE ADDRESS OF SUBROUTINE ADDRESS
                    *
     F5D1 A7 00               STA A   0,X          ;SAVE VECTOR ADDRESS,HIGH
     F5D3 E7 01               STA B   1,X          ;SAVE VECTOR ADDRESS,LOW
     F5D5 EE 00               LDX     0,X          ;LOAD VECTOR ADDRESS INTO X
     F5D7 EB 01               ADD B   1,X
```

```
SEQ  LOC  OBJ      SOURCE

     F5D9 A9 00              ADC A   0,X
                   *  ADD IN OFFSET CONTAINED IN VECTOR TABLE
     F5DB 30                 TSX
     F5DC A7 00              STA A   0,X
     F5DE E7 01              STA B   1,X
     F5E0 A6 02              LDA A   2,X
     F5E2 06                 TAP                 ;STORE OLD STATE INTO CC
     F5E3 EE 00              LDX     0,X
                   *
     F5E5 31                 INS
     F5E6 31                 INS                 ;CORRECT SP
                   *  JUMP  TO SUBROUTINE
     F5E7 AD 00              JSR     0,X
                   *
                   *  NORMAL EXIT FROM SUBROUTINE
                   *   INCREMENT RETURN ADDRESS
     F5E9 30                 TSX
     F5EA 6C 06              INC     6,X
     F5EC 26 02              BNE     *+4
     F5EE 6C 05              INC     5,X
                   *  EXIT
     F5F0 3B                 RTI
                   *
                   *  STACK ELEMENTS ARE STACK POINTER +2 SINCE JSR
                   *
               0002 UC       EQU     2           ;CC RELATIVE TO SP
               0003 UB       EQU     3           ;B  RELATIVE TO SP
               0004 UA       EQU     4           ;A  RELATIVE TO SP
               0005 UXH      EQU     5           ;XH RELATIVE TO SP
               0006 UXL      EQU     6           ;XL RELATIVE TO SP
               0007 URH      EQU     7           ;RH RELATIVE TO SP
               0008 URL      EQU     8           ;RL RELATIVE TO SP
                   *
                   *  PUSH ALL ONTO STACK -- REGISTERS CORRECT ON EXIT
                   *
               0000 SRH      EQU     0           ;SYSTEM RETURN H RELATIVE TO SP
               0001 SRL      EQU     1           ;SYSTEM RETURN L RELATIVE TO SP
                   *
                   *  PUSH ALL REGISTERS ONTO STACK--REGISTERS UNMODIFIED
                   *
                   *  CURRENT STACK SP  +1  +2  +3  +4  +5  +6  +7  +8  +9
                   *                   SRH SRL  CC  B   A   XH  XL  URL URH
                   *  RESULT STACK BEFORE RETURN TO MAIN EXIT
                   *SRH SRL CC B A XH XL URL URH CC B A XH XL
                   *
               F5F1 PUSHAL   EQU     *
                   *
                   *  MAKE SPACE
                   *
     F5F1 34                 DES
     F5F2 34                 DES
     F5F3 34                 DES
     F5F4 34                 DES
     F5F5 34                 DES
                   *
```

```
SEQ  LOC  OBJ      SOURCE

1211               *  MOVE STACK DOWN
1212               *
1213 F5F6 C6 09          LDA B   #9            ;NINE BYTES DOWN
1214 F5F8 30             TSX
1215 F5F9 A6 05    AS    LDA A   5,X           ;OFFSET OF 5
1216 F5FB A7 00          STA A   0,X
1217 F5FD 08             INX
1218 F5FE 5A             DEC B
1219 F5FF 26 F8          BNE     AS
1220               *
1221               *  RECOPY "PUSHED" REGISTERS
1222               *
1223 F601 C6 05          LDA B   #5            ;FIVE BYTES TO MOVE
1224 F603 30             TSX
1225 F604 A6 02    AC    LDA A   UC,X
1226 F606 A7 09          STA A   UC+7,X        ;OFFSET BY 7
1227 F608 08             INX
1228 F609 5A             DEC B
1229 F60A 26 F8          BNE     AC
1230               *
1231               *  EXIT TO MAIN
1232               *
1233 F60C 39             RTS
1234               *
1235               *  USER STACK IS
1236               *                  CC  B  A  XH  XL
1237               *               SP
1238               *
1239               *  POP ALL REGISTERS
1240               *
1241          F60D POPALL  EQU     *
1242 F60D 30             TSX
1243               *  CURRENT STACK
1244               *  SRH  SRL  CC  B  A  XH  XL  URH  URL  CC  B  A  XH  XL
1245               *  RESULT STACK
1246               *                         SRH  SRL  CC  B  A  XH  XL  URH  URL
1247               *
1248               *  RECOPY "PULLED" REGISTERS
1249               *
1250 F60E C6 05          LDA B   #5            ;FIVE OF THEM
1251 F610 A6 09    PC    LDA A   UC+7,X        ;OFFSET BY 7
1252 F612 A7 02          STA A   UC,X
1253 F614 08             INX
1254 F615 5A             DEC B
1255 F616 26 F8          BNE     PC
1256               *
1257               *  SHIFT EVERYTHING OVER
1258               *
1259 F618 C6 09          LDA B   #9            ;NINE BYTES
1260 F61A A6 03    PS    LDA A   URL-5,X
1261 F61C A7 08          STA A   URL,X         ;OFFSET 5
1262 F61E 09             DEX
1263 F61F 5A             DEC B
1264 F620 26 F8          BNE     PS
1265               *
```

```
SEQ  LOC  OBJ      SOURCE

1266                *  FINALLY INCREMENT SP
1267                *
1268 F622 31                INS
1269 F623 31                INS
1270 F624 31                INS
1271 F625 31                INS
1272 F626 31                INS
1273 F627 39                RTS
1274                *
1275                *  TRANSFER X TO A,B
1276                *
1277 F628 30        TXAB    TSX
1278 F629 A6 05             LDA A  UXH,X        ;X HIGH
1279 F62B E6 06             LDA B  UXL,X        ;X LOW
1280 F62D A7 04     STAB    STA A  UA,X         ;TO A
1281 F62F E7 03             STA B  UB,X         ;TO B
1282                *
1283 F631 39                RTS
1284                *
1285                *  TRANSFER A,B TO X
1286                *
1287 F632 30        TABX    TSX
1288 F633 A6 04             LDA A  UA,X         ;A
1289 F635 A7 05             STA A  UXH,X        ;TO X HIGH
1290                *
1291 F637 A6 03             LDA A  UB,X         ;B
1292 F639 E7 06             STA B  UXL,X        ;TO X LOW
1293                *
1294 F63B 39                RTS
1295                *
1296                *  EXCHANGE A,B AND X
1297                *
1298           F63C XABX    EQU    *
1299 F63C 30                TSX
1300                *   CURRENT STACK
1301                *              SRH  SRL  C  B  A  XH  XL  URH  URL
1302                *   RESULT                     XL  XH  A  B
1303                *
1304 F63D A6 05             LDA A  UXH,X        ;PICK UP UX
1305 F63F 36                PSH A
1306 F640 E6 06             LDA B  UXL,X
1307 F642 8D EF             BSR    TABX+1       ;THEN GO TO XFER A,B TO X
1308 F644 32                PUL A
1309 F645 20 E6             BRA    STAB         ;TO STORE IN A,B
1310                *
1311                *  PUSH X
1312                *
1313           F647 PUSX    EQU    *
1314                *
1315                *  GET SPACE IN SPACE
1316                *
1317 F647 34                DES
1318 F648 34                DES
1319 F649 30                TSX
1320                *  MOVE STACK DOWN TWO
```

```
LOC  OBJ      SOURCE

F64A 86 09              LDA A  #9           ;MOVE TOTAL OF 9 BYTES
                *
F64C E6 02      SA      LDA B  2,X
F64E E7 00              STA B  0,X
F650 08                 INX
F651 4A                 DEC A
F652 26 F8              BNE    SA
                *
                *  STACK MOVED -- INSERT X
F654 30                 TSX
F655 A6 05              LDA A  UXH,X
F657 A7 09              STA A  UXH+4,X
F659 A6 06              LDA A  UXL,X
F65B A7 0A              STA A  UXL+4,X
                *
F65D 39                 RTS
                *   STACK ON RETURN
                *       SRH  SRL  C  B  A  XH  XL  URH  URL  XH  XL
                *    SP
                *
                *  PULL X
           F65E PULX    EQU    *
                *  GET X FROM STACK
                *
F65E 30                 TSX
F65F A6 09              LDA A  UXH+4,X      ;CURRENT X ON STACK
F661 A7 05              STA A  UXH,X        ;REG X
F663 A6 0A              LDA A  UXL+4,X
F665 A7 06              STA A  UXL,X
                *
                *  NOW MOVE UP TWO
F667 86 09              LDA A  #9           ;BYTE COUNT
           F669 PA      EQU    *
F669 E6 08              LDA B  8,X
F66B E7 0A              STA B  10,X
F66D 09                 DEX
F66E 4A                 DEC A
F66F 26 F8              BNE    PA
                *  UPDATE SP
F671 31                 INS
F672 31                 INS
                *
F673 39                 RTS
                *
                *  ADD X TO A,B
                *
           F674 ADDXAB  EQU    *
F674 30                 TSX
F675 8D C6              BSR    XABX+1       ;EASY WAY EXCHANGE AB & X
F677 8D 03              BSR    ADDABX+1     ;ADD OTHER WAY
F679 20 C2              BRA    XABX+1       ;THEN EXCHANGE BACK
                *
                *  ADD A,B TO X
                *
F67B 30         ADDABX  TSX
```

```
LOC  OBJ      SOURCE

F67C A6 03              LDA A   UB,X
F67E E6 04              LDA B   UA,X
                 *
                 *  CODE SHARED BY ADDAX,INDEX
                 *
          F680   ADDAB  EQU     *
F680 AB 06              ADD A   UXL,X          ;ADD UXL TO UB
F682 A7 06              STA A   UXL,X          ;STORE INTO UXL
                 *
F684 E9 05              ADC B   UXH,X          ;ADD UXH TO UA
F686 07          STAUXH TPA                    ;SAVE STATUS
F687 E7 05              STA B   UXH,X          ;STORE IN UXH
                 *
F689 6D 06              TST     UXL,X          ;TEST LOW BYTE FOR ZERO
                 *
                 *  CODE SHARED BY ADDABX,MUL8,MUL16
                 *
          F68B   TESTZ  EQU     *
F68B 27 02              BEQ     TA             ;HIGH BY STAT IS TRUE RSLT
F68D 84 FB              AND A   #$FB           ;NO,Z BIT IS CLEARED
F68F A7 02       TA     STA A   UC,X           ;SAVE STATUS
                 *
F691 39                 RTS
                 *
                 *  ADD A TO X
                 *
          F692   ADDAX  EQU     *
F692 30                 TSX
F693 A6 04              LDA A   UA,X
          F695   ADDZ   EQU     *
F695 C6 00              LDA B   #0
F697 20 E7              BRA     ADDAB
                 *  ADD B TO X
          F699   ADDBX  EQU     *
F699 30                 TSX
F69A A6 03              LDA A   UB,X
F69C 20 F7              BRA     ADDZ
                 *
                 *  SUBTRACT X FROM A,B
                 *
F69E 30          SUBXAB TSX
F69F 8D 9C              BSR     XABX+1
F6A1 8D 03              BSR     SUBABX+1
F6A3 20 98              BRA     XABX+1
                 *
                 *  SUBTRACT A,B FROM X
                 *
          F6A5   SUBABX EQU     *
F6A5 30                 TSX
F6A6 E6 05              LDA B   UXH,X
F6A8 A6 06              LDA A   UXL,X
                 *
F6AA A0 03              SUB A   UB,X
F6AC A7 06              STA A   UXL,X
                 *
```

```
SEQ  LOC  OBJ      SOURCE

1431 F6AE E2 04              SBC B   UA,X
1432 F6B0 20 D4              BRA     STAUXH
1433                *
1434                *  SUBTRACT A FORM X
1435                *
1436           F6B2 SUBAX    EQU     *
1437 F6B2 30                 TSX
1438 F6B3 E6 04              LDA B   UA,X
1439 F6B5 A6 06    SSUB      LDA A   UXL,X          ;SUB A FROM XL
1440 F6B7 10                 SBA
1441 F6B8 A7 06              STA A   UXL,X          ;STORE     XL
1442                *
1443 F6BA E6 05              LDA B   UXH,X
1444 F6BC C2 00              SBC B   #0
1445 F6BE 20 C6              BRA     STAUXH
1446                *
1447                *  SUB B FROM X
1448                *
1449           F6C0 SUBBX    EQU     *
1450 F6C0 30                 TSX
1451 F6C1 E6 03              LDA B   UB,X
1452 F6C3 20 F0              BRA     SSUB
1453                *
1454                *  INDEX: X:=X+A+B  (SAVE USER A,B)
1455                *
1456           F6C5 INDEX    EQU     *
1457 F6C5 8D 11              BSR     MPY8           ;A,B := USERA+USERB
1458                *
1459                *  EXCHANGE A&B TO SHARE CODE WITH ADDABX
1460                *
1461 F6C7 37                 PSH B
1462 F6C8 16                 TAB
1463 F6C9 32                 PUL A
1464 F6CA 30                 TSX
1465 F6CB 20 B3              BRA     ADDAB
1466                *
1467                *  MUL8: A,B := A+B
1468                *
1469           F6CD MUL8     EQU     *
1470 F6CD 8D 09              BSR     MPY8
1471 F6CF 30                 TSX
1472 F6D0 E7 03              STA B   UB,X           ;SAVE RESULT
1473 F6D2 A7 04              STA A   UA,X           ;SET UP N BIT
1474 F6D4 07                 TPA
1475 F6D5 5D                 TST B
1476 F6D6 20 B3    JMPTZ     BRA     TESTZ          ;UPDATE USER C & RETURN
1477           0800          ORG     $0800
1478                * NOT INCLUDED IN MONITOR
1479                *  MUL16--16 BIT MULTIPLY
1480                *    A,B,X := A,B+X
1481                *    A,B = PARTIAL PRODUCT
1482                *    USERX = MULTIPLIER
1483                *    USERA,USERB = MULTIPLICAND
1484                *
1485 0800 86 10    MUL16     LDA A   #16            ;PUSH COUNTER INTO STACK
```

```
SEQ  LOC  OBJ      SOURCE

1486 0802 36                 PSH A
1487 0803 30                 TSX
1488 0804 4F                 CLR A
1489 0805 5F                 CLR B
1490 0806 66 06              ROR     UXH+1,X       ;SHIFT LSB INTO CRY
1491 0808 66 07              ROR     UXL+1,X
1492                   *
1493                   *  LOOP 16 TIMES .
1494                   *
1495 080A 24 04        MLOOP   BCC     MSHIFT        ;MULTIPLIER IS EVEN
1496 080C EB 04              ADD B   UB+1,X        ;A,B := A,B+USERA,B
1497 080E A9 05              ADC A   UA+1,X
1498                   *
1499 0810 46           MSHIFT  ROR A                 ;SHIFT EVERYTHING RIGHT
1500 0811 56                 ROR B
1501 0812 66 06              ROR     UXH+1,X
1502 0814 66 07              ROR     UXL+1,X
1503 0816 6A 00              DEC     0,X           ;DEC COUNTER
1504 0818 26 F0              BNE     MLOOP
1505                   *
1506                   *  END LOOP
1507                   *
1508 081A 31                 INS                   ;RESTORE SP
1509 081B 30                 TSX
1510 081C E7 03              STA B   UB,X
1511 081E A7 04              STA A   UA,X
1512                   *
1513                   *  SET USER CC:= N=N(MSBYTE)
1514                   *     Z:=AND (Z(MSBYTE),,,,,Z(LSBYTE)):
1515                   *     V:= 0
1516                   *     CRY := 0.
1517                   *     THE LAST ADD RTESET CRY.  STA SET N=N(MSBYTE) & V=0
1518                   *
1519 0820 07                 TPA
1520 0821 EA 05              ORA B   UXH,X         ;B:= OR OF LAST 3 LS BYTES
1521 0823 EA 06              ORA B   UXL,X
1522                   *
1523                   *  USER CC HAS CORRECT N,V,&C.  CC HAS CORRECT Z FOR LS BYTE
1524                   *  GO TO END OF ADDXAB TO UPDATE REGISTERS
1525                   *
1526 0825 7E F6D6            JMP     JMPTZ
1527           F6D8          ORG     $F6D8
1528                   *
1529                   *  SUBROUTINE MPY8.  A,B:= USERA*USERB
1530                   *      A = PARTIAL PRODUCT
1531                   *      B = MULTIPLIER & LSB'S OF PAR.  PROD.
1532                   *      USERA = MULTIPLICAND
1533                   *
1534 F6D8 86 08        MPY8    LDA A   #8            ;PUSH COUNTER ONTO STACK
1535 F6DA 36                 PSH A
1536                   *  STACK = COUNT,R,R,R,R,C,B,A,X,X,R,R
1537 F6DB 4F                 CLR A
1538 F6DC 30                 TSX
1539 F6DD E6 06              LDA B   UB+3,X        ;B = MULTIPLIER
1540 F6DF 56                 ROR B
```

```
SEQ  LOC  OBJ       SOURCE

1541                *
1542                *  LOOP 8 TIMES
1543                *
1544 F6E0 24 02     BLOOP    BCC     BSHIFT       ;MULTIPLIER IS EVEN
1545 F6E2 AB 07              ADD A   UA+3,X
1546                *
1547 F6E4 46        BSHIFT   ROR A                ;SHIFT LSB OF A INTO B
1548 F6E5 56                 ROR B
1549 F6E6 6A 00              DEC     0,X          ;CHECK COUNT
1550 F6E8 26 F6              BNE     BLOOP
1551                *
1552                *  END OF LOOP
1553                *
1554 F6EA 31                 INS                  ;RESTORE SP
1555 F6EB 39                 RTS
1556                *
1557                *  RELATIVE ENTRY POINTS TO SUBROUTINE VECTOR
1558                *
1559           F6EC SVECTO   EQU     *
1560                *                            INDEX
1561 F6EC FF 05              FDB     $FF05        ;PUSHALL-*   0
1562 F6EE FF 1F              FDB     $FF1F        ;POPALL-*    1
1563 F6F0 FF 38              FDB     $FF38        ;TXAB-*      2
1564 F6F2 FF 40              FDB     $FF40        ;TABX-*      3
1565 F6F4 FF 48              FDB     $FF48        ;XABX-*      4
1566 F6F6 FF 51              FDB     $FF51        ;PUSX-*      5
1567 F6F8 FF 66              FDB     $FF66        ;PULX-*      6
1568 F6FA FF 7A              FDB     $FF7A        ;ADDXAB-*    7
1569 F6FC FF 7F              FDB     $FF7F        ;ADDABX-*    8
1570 F6FE FF 94              FDB     $FF94        ;ADDAX-*     9
1571 F700 FF 99              FDB     $FF99        ;ADDBX-*     10
1572 F702 FF 9C              FDB     $FF9C        ;SUBXAB-*    11
1573 F704 FF A1              FDB     $FFA1        ;SUBABX-*    12
1574 F706 FF AC              FDB     $FFAC        ;SUBAX-*     13
1575 F708 FF B8              FDB     $FFB8        ;SUBBX-*     14
1576 F70A 00 17              FDB     $0017        ;P2HEX-*     15
1577 F70C 00 10              FDB     $0010        ;P4HEX-*     16
1578 F70E 00 2F              FDB     $002F        ;PRINTA-*    17
1579 F710 00 4F              FDB     $004F        ;PMESS-*     18
1580 F712 00 5D              FDB     $005D        ;VALAN-*     19
1581 F714 00 86              FDB     $0086        ;INPUTA-*    20
1582 F716 00 96              FDB     $0096        ;CONHB-*     21
1583 F718 FF AD              FDB     $FFAD        ;INDEX-*     22
1584 F71A FF B3              FDB     $FFB3        ;MUL8-*      23
1585                *
1586                *  PRINT 2/4 HEX CHARS FROM MEM(UX),UX+1
1587                *     UX IS INCREMENTED UPON OUTPUT I.E. UX = UX+2
1588                *
1589           F71C P4HEX    EQU     *
1590 F71C 30                 TSX
1591 F71D EE 05              LDX     UXH,X        ;USERS X
1592 F71F 8D 06              BSR     PHEX         ;PRINT MEM()
1593                *
1594                *  PRINT 2 HEX CHARS FROM MEM(UX)
1595           F721 P2HEX    EQU     *
```

```
LOC  OBJ          SOURCE

F721 30                      TSX
F722 EE 05                   LDX    UXH,X        ;USERS X
F724 8D 01                   BSR    PHEX         ;PRINT MEM(X)
F726 39                      RTS
                  *
                  *  PRINT 2 HEX CHARS FROM MEM(X)
                  *
          F727 PHEX     EQU    *
F727 A6 00                   LDA A  0,X          ;GET THE CHAR
F729 8D 29                   BSR    ASCIIR       ;CONV RIGHT NIB & RSLT IN A
F72B 36                      PSH A               ;SAVE IT
F72C A6 00                   LDA A  0,X          ;GET CHAR AGAIN
F72E 8D 20                   BSR    ASCIIL       ;CONV THE LEFT NIBBLE INTO A
F730 8D 0E                   BSR    PUTAX        ;PRINT A REG CHAR
F732 32                      PUL A               ;RECOVER SAVED
F733 8D 10                   BSR    PUTA         ;..THEN FALL INTO PINCX
                  *
                  *  INCREMENT THE USER'S X IN THE STACK
                  *
          F735 PINCX    EQU    *
F735 30                      TSX                 ;SP IS +2 SINCE 2 BSR DOWN IN CA  S
F736 6C 08                   INC    UXL+2,X      ;INC MEMORY X LOW
F738 26 02                   BNE    PRTS         ;OVERFLOW MEANS INC HIGH PART
F73A 6C 07                   INC    UXH+2,X      ;YES--INC HIGH
F73C 39           PRTS       RTS                 ;EXIT
                  *
                  *  PRINT THE CHAR IN USERS A
                  *
          F73D PRINTA   EQU    *
F73D 30                      TSX
F73E A6 04                   LDA A  UA,X         ;GET THE CHAR
                  *
                  *  PRINT CHAR IN DESIGNATED REG
                  *  ACIA ADDRESS IN X
                  *
                  *  PRINT CHAR IN A
F740 FE FFF6 PUTAX           LDX    ACIAI        ;GET INDIR ADDR OF ACIA
F743 EE 00                   LDX    0,X          ;GET ACTUAL ADD OF ACIA IN X
F745 36           PUTA       PSH A               ;SAVE REGISTER
F746 A6 00        PRDY       LDA A  0,X          ;ACIA STATUS
F748 85 02                   BIT A  #02          ;READY?
F74A 27 FA                   BEQ    PRDY         ;NOT READY
F74C 32                      PUL A               ;RESTORE CHAR
F74D A7 01                   STA A  1,X          ;PRINT CHAR
F74F 39                      RTS
                  *
                  *  CONVERT A FROM BINARY  TO HEXIN LEFT/RIGHT NIBBLE
                  *   LEFT PART
                  *
          F750 ASCIIL   EQU    *
F750 44                      LSR A               ;A HAS CHAR TO BE CONVERTED
F751 44                      LSR A
F752 44                      LSR A
F753 44                      LSR A
                  *  IN POSITION
```

```
SEQ  LOC  OBJ      SOURCE

1651 F754 84 0F    ASCIIR   AND A   #$0F         ;CLEAR LEFT PART
1652 F756 8B 30             ADD A   #$30
1653 F758 81 39             CMP A   #$39         ;0 TO 9
1654 F75A 23 02             BLS     ARTS         ;YES DONE
1655 F75C 8B 07             ADD A   #7           ;NO THEN A TO F
1656 F75E 39       ARTS     RTS
1657                *
1658                *  PRINT MESSAGE POINTED TO BY X AND TERMINATED BY ETX
1659                *
1660           F75F PMESS   EQU     *
1661           0004 ETX     EQU     $04
1662 F75F 30                TSX
1663 F760 EE 05             LDX     UXH,X        ;GET USERS X
1664 F762 A6 00             LDA A   0,X          ;GET CHAR
1665 F764 81 04             CMP A   #ETX         ;IS IT TERMINATOR?
1666 F766 27 06             BEQ     TRTS         ;DONE
1667 F768 8D D6             BSR     PUTAX        ;PRINT A
1668 F76A 8D C9             BSR     PINCX        ;INC USERS X
1669 F76C 20 F1             BRA     PMESS        ;LOOP TILL DONE
1670 F76E 39       TRTS     RTS
1671                *
1672                *  X HAS ADDRESS OF CHAR TO BE TESTED
1673                *     FOR BEING ALPHA NUMERIC
1674                *     CARRY SET IF TRUE
1675           F76F VALAN   EQU     *
1676 F76F 30                TSX
1677 F770 EE 05             LDX     UXH,X        ;GET CHAR ADDR
1678 F772 8D 05             BSR     ALPNUM       ;TEST MEM(X)=ALPHANUMERIC
1679                *
1680                *  SET USER'S CARRY = CURRENT CARRY (AND OTHER FLAGS)
1681                *
1682 F774 07       SCARRY   TPA
1683                *
1684 F775 30       SETUS    TSX
1685 F776 A7 02             STA A   UC,X
1686 F778 39                RTS
1687                *
1688                *  SET CARRY IF MEM(X) IS ALPHANUMERIC
1689                *    CLEAR V IF HEX DIGIT
1690                *
1691           F779 ALPNUM  EQU     *
1692 F779 A6 00             LDA A   0,X          ;GET THE CHAR
1693 F77B 81 41             CMP A   #'A
1694 F77D 2D 0E             BLT     ANUM         ;TOO SMALL FOR ALPHA,= NUM
1695 F77F 81 5A             CMP A   #'Z
1696 F781 2E 12             BGT     ANOTOK       ;BIGGER THAN "Z"
1697 F783 81 C7             CMP A   #$C7         ;SET V IF >F
1698 F785 29 10             BVS     NRTS         ;QUIT IF NOT HEX (C=1)
1699 F787 80 07             SUB A   #7           ;CONVERT LETTER TO HEX
1700 F789 84 0F    AOK      AND A   #15          ;STRIP OVERBITS FROM HEX DIGIT
1701 F78B 0D                SEC                  ;SET C FOR VALID A/N
1702 F78C 39                RTS
1703                *
1704 F78D 81 30    ANUM     CMP A   #'0          ;NUMERIC TESTING
1705 F78F 2D 04             BLT     ANOTOK       ;NOT NUMERIC
```

```
SEQ  LOC  OBJ      SOURCE

     F791 81 39              CMP A   #'9
     F793 2F F4              BLE     AOK          ;IT IS IN 0-9
     F795 0C        ANOTOK   CLC                  ;RESET CARRY FOR NOT A/N
     F796 0B                 SEV                  ;SET V FOR NOT HEX EITHER
     F797 39        NRTS     RTS
                    *
     F798 20 9B     JPINCX   BRA     PINCX        ;EXTRA BRA TO REACH PINCX
                    *
                    *  INPUTA:
                    *    INPUT ACIA DATA INTO A REG
                    *    STRIP PARITY
                    *
          F79A INPUTA        EQU     *
     F79A FE FFF6            LDX     ACIAI        ;GET INDIRECT ADDRESS
     F79D EE 00              LDX     0,X          ;GET ACIA ADDRESS
                    *
     F79F A6 00     IWAIT    LDA A   0,X          ;ACIA STATUS
     F7A1 47                 ASR A                ;CARRY:= RDRF
     F7A2 24 FB              BCC     IWAIT        ;NO INPUT,LOOP
                    *
     F7A4 A6 01              LDA A   1,X          ;ACIA DATA
     F7A6 84 7F              AND A   #$7F         ;STRIP PARITY
     F7A8 30                 TSX                  ;PUT RESULT ONTO STACK
     F7A9 A7 04              STA A   UA,X
     F7AB 39                 RTS
                    *
                    *   CONHB--CONVERT HEX TO BINARY:
                    *     SCAN UP TO B ASCII CHARS STARTING AT X
                    *     LOOKING FOR A VALID HEX NUMBER. RETURN BINARY
                    *     EQUIVALENT OF NUMBER IN A,B.  IF NUMBER HAS MORE
                    *     THAN 16 BITS,IGNORE MSB'S.
                    *
                    *   INPUT: X= ADDRESS OF 1ST CHAR TO BE SCANNED
                    *          B= MAX. # OF CHARS TO BE SCANNED
                    *
                    *   OUTPUT: A,B=BINARY RESULT
                    *           CARRY=1 IF VALID NUMBER IS FOUND
                    *           X POINTS TO LAST CHAR SCANNED
                    *
          F7AC CONHB         EQU     *
     F7AC 30                 TSX
     F7AD E6 03              LDA B   UB,X         ;GET MAX COUNT
     F7AF 6F 04              CLR     UA,X         ;CLEAR USERS A,B REGS
     F7B1 6F 03              CLR     UB,X
                    *
                    *  LOOP WHILE NOT ALPHANUMERIC AND COUNT > 1
                    *
     F7B3 30        CLOOP1   TSX
     F7B4 EE 05              LDX     UXH,X        ;GET CHAR ADDRESS
     F7B6 8D C1              BSR     ALPNUM       ;IS MEM(X) A/N?
     F7B8 25 09              BCS     CFOUND       ;YES,STOP SCANNING
     F7BA 5A                 DEC B                ;DEC COUNT
     F7BB 2F 04              BLE     ENDCNT       ;COUNT EXHAUSTED
     F7BD 8D D9              BSR     JPINCX       ;INC USERS X
     F7BF 20 F2              BRA     CLOOP1
```

```
SEQ  LOC  OBJ      SOURCE

1761                *
1762                *  END LOOP
1763                *
1764                *  COUNT EXHAUSTED WITH NO SUCCESS.
1765                *  (CARRY WAS RESET BY ALPNUM)
1766                *
1767 F7C1 20 B1     ENDCNT  BRA   SCARRY      ;RESET USER C AND RETURN
1768                *
1769                *  WHILE HEX AND COUNT >0 SHIFT MEM(X) INTO UA,UB
1770                *
1771                *  BEGIN OUTER LOOP
1772 F7C3 30        CFOUND  TSX
1773 F7C4 EE 05             LDX   UXH,X
1774 F7C6 8D B1             BSR   ALPNUM      ;CNVT MEM(X) TO HEX
1775 F7C8 29 18             BVS   NOGOOD      ;INVALID CHAR
1776 F7CA 37                PSH B             ;SAVE COUNT
1777 F7CB C6 04             LDA B #4          ;LOOP COUNT
1778                *
1779                *  SHIFT LEFT UA,UB
1780                *
1781 F7CD 30                TSX
1782 F7CE 68 04     CSLOOP  ASL   UB+1,X      ;+1 TO COMP.FOR PUSH
1783 F7D0 69 05             ROL   UA+1,X
1784 F7D2 5A                DEC B
1785 F7D3 2E F9             BGT   CSLOOP
1786                *
1787 F7D5 AA 04             ORA A UB+1,X      ;'OR' ON NEW CHAR
1788 F7D7 A7 04             STA A UB+1,X
1789 F7D9 33                PUL B             ;RETRIEVE COUNT
1790 F7DA 8D BC             BSR   JPINCX      ;INC USER X
1791 F7DC 5A                DEC B
1792 F7DD 2E E4             BGT   CFOUND      ;REPEAT
1793                *  END OF OUTER LOOP
1794 F7DF 0D                SEC               ;VALID NUMBER
1795 F7E0 20 92             BRA   SCARRY      ;SET USER C AND RETURN
1796                *
1797                *  NON-HEX CHAR FOUND. IF CHAR=G-Z,THIS IS NOT A VALID
1798                *  HEX NUMBER  OTHERWISE, CHAR IS A DELIMITER AND
1799                *  NUMBER IS VALID.
1800                *
1801 F7E2 07        NOGOOD  TPA               ;TOGGLE CARRY BIT
1802 F7E3 4C                INC A
1803 F7E4 20 8F             BRA   SETUS       ;SETUP USER STATUS & RETURN
1804                *
1805                        END

1805 LINES ASSEMBLED,  LOC = F7E6,  0000 ERRORS DETECTED.
```

SYMBOL	VALUE	ATTR	LOCN	LINK
ACIAC	6000	84	2AA2	
ACIAD	6001	84	2AAB	
ACIAS	6000	84	2AB4	
BLANK	0020	84	2ABD	
CR	000D	84	2AC6	
ESC	001B	84	2ACF	
EOT	0004	84	2AD8	
LAST	FFFF	84	2AE1	
LF	000A	84	2AEA	
RUBOUT	007F	84	2AF3	
MONFG	0051	84	2AFC	
NORFG	00E6	84	2B05	
VSFLG	00E7	84	2B0E	
BASE	FF90	84	2B17	
BOS	FF8F	84	2B20	
BUF	FF90	84	2B29	
PROMAD	FFD8	84	2B32	
OFFSET	FFD8	84	2B3B	
ADR	FFDA	84	2B44	
ADDL	FFDC	84	2B4D	
ADDH	FFDE	84	2B56	
BUFPTR	FFE0	84	2B5F	
RECTYP	FFE2	84	2B68	
COUNT	FFE3	84	2B71	
CKSM	FFE4	84	2B7A	
SAVESP	FFE5	84	2B83	
SAVEX	FFE7	84	2B8C	
ECHO	FFE9	84	2B95	
TCOUNT	FFEA	84	2B9E	
CREG	FFEB	84	2BA7	
BREG	FFEC	84	2BB0	
AREG	FFED	84	2BB9	
XREG	FFEE	84	2BC2	
PREG	FFF0	84	2BCB	
SREG	FFF2	84	2BD4	
USWI	FFF4	84	2BDD	
ACIAI	FFF6	84	2BE6	
IRQVEC	FFF8	84	2BEF	
SWIVEC	FFFA	84	2BF8	
NMIVEC	FFFC	84	2C01	
PWRI	F51C	84	2C0A	
START	F018	84	2C13	
START1	F01F	84	2C1C	
BREAK	F01A	84	2C25	
BREAK1	F0CF	84	2C2E	
ACIAA	F01D	84	2C37	
SWI30	F0E9	84	2C40	
SWIHAN	F0D6	84	2C49	
MONENT	F059	84	2C52	
MONEN1	F059	84	2C5B	
FINIT	F434	84	2C64	
PCRLF	F322	84	2C6D	
MONITR	F05F	84	2C76	
RDROFF	F3B9	84	2C7F	

SYMBOL	VALUE	ATTR	LOCN	LINK
OUTCH	F2FB	84	2C88	
RT10	F077	84	2C91	
RT20	F07E	84	2C9A	
ABORT	F0C5	84	2CA3	
WAITTY	F41E	84	2CAC	
RT30	F081	84	2CB5	
RT90	F084	84	2CBE	
PXISTS	F3A3	84	2CC7	
CTABLE	F0A4	84	2CD0	
DLOOP	F092	84	2CD9	
DL10	F09A	84	2CE2	
CTEND	F0C5	84	2CEB	
MONEND	F05F	84	2CF4	
LOAD	F1CB	84	2CFD	
GO	F1B7	84	2D06	
PUNCH	F326	84	2D0F	
PREGS	F106	84	2D18	
MOVE	F5A2	84	2D21	
VFY	F4D3	84	2D2A	
READ	F4E0	84	2D33	
SM	F401	84	2D3C	
DM	F14E	84	2D45	
EOF	F175	84	2D4E	
BADINP	F0C5	84	2D57	
MQUES	F291	84	2D60	
MSGMON	F0C8	84	2D69	
MSGABT	F0C8	84	2D72	
SWI40	F0F0	84	2D7B	
SWI20	F0E4	84	2D84	
RSRSR	F5BE	84	2D8D	
SWI50	F0F3	84	2D96	
PR1	F10B	84	2D9F	
RESTAK	F124	84	2DA8	
PR10	F110	84	2DB1	
PSPACE	F39E	84	2DBA	
P4HEXS	F39A	84	2DC3	
PR20	F11A	84	2DCC	
RUS10	F12A	84	2DD5	
CHEKSM	F134	84	2DDE	
NEXT2D	F2BC	84	2DE7	
CS1	F141	84	2DF0	
MCSER	F296	84	2DF9	
GETRNG	F185	84	2E02	
DM10	F150	84	2E0B	
DM20	F156	84	2E14	
DM50	F172	84	2E1D	
MPEOF	F2A8	84	2E26	
NULLS	F17A	84	2E2F	
NULL1	F17C	84	2E38	
NXTADR	F2D6	84	2E41	
GETRG3	F19C	84	2E4A	
GETRG1	F196	84	2E53	
GETRG4	F1AF	84	2E5C	
MRNGER	F283	84	2E65	

SYMBOL	VALUE	ATTR	LOCN	LINK
RNGERR	F1A9	84	2E6E	
G10	F1C8	84	2E77	
LOOFST	F1D4	84	2E80	
LHF2	F1FB	84	2E89	
RDRON	F3C3	84	2E92	
FINDS	F270	84	2E9B	
RDPRE	F1FE	84	2EA4	
LHF3	F231	84	2EAD	
LHF4	F249	84	2EB6	
LDR10	F235	84	2EBF	
SETOFF	F3CD	84	2EC8	
LHF9	F24D	84	2ED1	
BADTAP	F260	84	2EDA	
MEOF	F28D	84	2EE3	
MTAPER	F29F	84	2EEC	
BT1	F26B	84	2EF5	
FS10	F273	84	2EFE	
MBADR	F27B	84	2F07	
MCRLFS	F2B3	84	2F10	
NA1	F2E2	84	2F19	
NA3	F2F6	84	2F22	
OUTCHX	F2F9	84	2F2B	
OC10	F30C	84	2F34	
OC20	F320	84	2F3D	
OCLOOP	F319	84	2F46	
PHF15	F339	84	2F4F	
PHF20	F339	84	2F58	
PUND10	F34B	84	2F61	
PUND20	F34D	84	2F6A	
PUNBYT	F392	84	2F73	
PREC10	F378	84	2F7C	
PXISTX	F3A6	84	2F85	
PX1	F3A6	84	2F8E	
PX2	F3B3	84	2F97	
RDF90	F3BB	84	2FA0	
RON90	F3C5	84	2FA9	
SETOUT	F3E2	84	2FB2	
SETPUL	F3E9	84	2FBB	
SETM1	F400	84	2FC4	
SETMEM	F3EA	84	2FCD	
PBADR	F3FA	84	2FD6	
SM5	F404	84	2FDF	
SM10	F40A	84	2FE8	
SM30	F41B	84	2FF1	
W10	F41E	84	2FFA	
W20	F427	84	3003	
W30	F433	84	300C	
MOVER	0001	84	3015	
DELAY	000A	84	301E	
PIA	5000	84	3027	
V50	0001	84	3030	
PROM	0004	84	3039	
RAM	FC00	84	3042	
P8BIN	F435	84	304B	

SYMBOL	VALUE	ATTR	LOCN	LINK
PSP	F448	84	3054	
A8	F43C	84	305D	
RASV	F44B	84	3066	
AA1	F465	84	306F	
AA3	F476	84	3078	
AA4	F491	84	3081	
VERR	F494	84	308A	
AT	F4B0	84	3093	
ADDRS	F4B7	84	309C	
VFY1	F505	84	30A5	
AV	F4D6	84	30AE	
AN	F4DC	84	30B7	
JVER	F516	84	30C0	
INCAD	F4EF	84	30C9	
AR	F4E3	84	30D2	
INK	F4F6	84	30DB	
EXIT	F502	84	30E4	
AX	F515	84	30ED	
BURN	F519	84	30F6	
OPMODE	D800	84	30FF	
MCINIT	F550	84	3108	
JBAD	F595	84	3111	
AM	F5AA	84	311A	
NITEMS	0018	84	3123	
LOCVV	F5C8	84	312C	
UC	0002	84	3135	
UB	0003	84	313E	
UA	0004	84	3147	
UXH	0005	84	3150	
UXL	0006	84	3159	
URH	0007	84	3162	
URL	0008	84	316B	
SRH	0000	84	3174	
SRL	0001	84	317D	
PUSHAL	F5F1	84	3186	
AS	F5F9	84	318F	
AC	F604	84	3198	
POPALL	F60D	84	31A1	
PC	F610	84	31AA	
PS	F61A	84	31B3	
TXAB	F628	84	31BC	
STAB	F62D	84	31C5	
TABX	F632	84	31CE	
XABX	F63C	84	31D7	
PUSX	F647	84	31E0	
SA	F64C	84	31E9	
PULX	F65E	84	31F2	
PA	F669	84	31FB	
ADDXAB	F674	84	3204	
ADDABX	F67B	84	320D	
ADDAB	F680	84	3216	
STAUXH	F686	84	321F	
TESTZ	F68B	84	3228	
TA	F68F	84	3231	

SYMBOL	VALUE	ATTR	LOCN	LINK
ADDAX	F692	84	323A	
ADDZ	F695	84	3243	
ADDBX	F699	84	324C	
SUBXAB	F69E	84	3255	
SUBABX	F6A5	84	325E	
SUBAX	F6B2	84	3267	
SSUB	F6B5	84	3270	
SUBBX	F6C0	84	3279	
INDEX	F6C5	84	3282	
MPY8	F6D8	84	328B	
MUL8	F6CD	84	3294	
JMPTZ	F6D6	84	329D	
MUL16	0800	84	32A6	
MSHIFT	0810	84	32AF	
MLOOP	080A	84	32B8	
BSHIFT	F6E4	84	32C1	
BLOOP	F6E0	84	32CA	
SVECTO	F6EC	84	32D3	
P4HEX	F71C	84	32DC	
PHEX	F727	84	32E5	
P2HEX	F721	84	32EE	
ASCIIR	F754	84	32F7	
ASCIIL	F750	84	3300	
PUTAX	F740	84	3309	
PUTA	F745	84	3312	
PINCX	F735	84	331B	
PRTS	F73C	84	3324	
PRINTA	F73D	84	332D	
PRDY	F746	84	3336	
ARTS	F75E	84	333F	
PMESS	F75F	84	3348	
ETX	0004	84	3351	
TRTS	F76E	84	335A	
VALAN	F76F	84	3363	
ALPNUM	F779	84	336C	
SCARRY	F774	84	3375	
SETUS	F775	84	337E	
ANUM	F78D	84	3387	
ANOTOK	F795	84	3390	
NRTS	F797	84	3399	
AOK	F789	84	33A2	
JPINCX	F798	84	33AB	
INPUTA	F79A	84	33B4	
IWAIT	F79F	84	33BD	
CONHB	F7AC	84	33C6	
CLOOP1	F7B3	84	33CF	
CFOUND	F7C3	84	33D8	
ENDCNT	F7C1	84	33E1	
NOGOOD	F7E2	84	33EA	
CSLOOP	F7CE	C4	33F3	

END ASM V1.1

```
SEQ  LOC  OBJ       SOURCE

0001                          NAM     FTBL
0002                *
0003                *GENERATED BY FTGN9*
0004                * * * * * * * * * * * *
0005                *FREQUENCY TABLES      *
0006                * * * * * * * * * * * *
0007           E800           ORG     $E800
0008                *
0009 E800 E8 80     INDEX     FDB     FR06
0010 E802 E8 D1               FDB     FR07
0011 E804 E9 1A               FDB     FR08
0012 E806 E9 5B               FDB     FR09
0013 E808 E9 98               FDB     FR010
0014 E80A E9 CD               FDB     FR011
0015 E80C E9 FE               FDB     FR012
0016 E80E EA 2F               FDB     FR013
0017 E810 EA 5C               FDB     FR014
0018 E812 EA 85               FDB     FR015
0019 E814 EA AE               FDB     FR016
0020 E816 EA D7               FDB     FR017
0021 E818 EA FC               FDB     FR018
0022 E81A EB 21               FDB     FR019
0023 E81C EB 42               FDB     FR020
0024 E81E EB 63               FDB     FR021
0025 E820 EB 84               FDB     FR022
0026 E822 EB A5               FDB     FR023
0027 E824 EB C6               FDB     FR024
0028 E826 EB E7               FDB     FR025
0029 E828 EC 04               FDB     FR026
0030 E82A EC 21               FDB     FR027
0031 E82C EC 3E               FDB     FR028
0032 E82E EC 5B               FDB     FR029
0033 E830 EC 78               FDB     FR030
0034 E832 EC 91               FDB     FR031
0035 E834 EC AA               FDB     FR032
0036 E836 EC C3               FDB     FR033
0037 E838 EC DC               FDB     FR034
0038 E83A EC F5               FDB     FR035
0039 E83C ED 0E               FDB     FR036
0040 E83E ED 27               FDB     FR037
0041 E840 ED 40               FDB     FR038
0042 E842 ED 59               FDB     FR039
0043 E844 ED 72               FDB     FR040
0044 E846 ED 8B               FDB     FR041
0045 E848 ED A4               FDB     FR042
0046 E84A ED B9               FDB     FR043
0047 E84C ED CE               FDB     FR044
0048 E84E ED E3               FDB     FR045
0049 E850 ED F8               FDB     FR046
0050 E852 EE 0D               FDB     FR047
0051 E854 EE 22               FDB     FR048
0052 E856 EE 37               FDB     FR049
0053 E858 EE 4C               FDB     FR050
0054 E85A EE 61               FDB     FR051
0055 E85C EE 76               FDB     FR052
```

```
SEQ  LOC  OBJ       SOURCE

0056 E85E EE 8B               FDB     FR053
0057 E860 EE A0               FDB     FR054
0058 E862 EE B5               FDB     FR055
0059 E864 EE CA               FDB     FR056
0060 E866 EE DB               FDB     FR057
0061 E868 EE EC               FDB     FR058
0062 E86A EE FD               FDB     FR059
0063 E86C EF 0E               FDB     FR060
0064            E880          ORG     $E880
0065                *FREQUENCY TABLE FOR 6 HZ.
0066                *
0067 E880 23        FR06      FCB     35          ;NO. -PULSE PERIODS IN 60 DEG SEG
0068 E881 03 1A               FDB     794           ;LENGTH OF PULSE PERIOD (USED
0069 E883 6C 8E               FDB     27790         ;LENGTH OF 60 DEGREE SEG (USE
0070 E885 07 FF               FDB     2047          ;SFMAX
0071 E887 00 80               FDB     128           ;SFMIN
0072                *PULSE WIDTHS
0073 E889 00 31               FDB     49          ;DOM
0074 E88B 00 30               FDB     48
0075 E88D 00 32               FDB     50          ;DOM
0076 E88F 00 2F               FDB     47
0077 E891 00 32               FDB     50          ;DOM
0078 E893 00 2E               FDB     46
0079 E895 00 33               FDB     51          ;DOM
0080 E897 00 2D               FDB     45
0081 E899 00 34               FDB     52          ;DOM
0082 E89B 00 2C               FDB     44
0083 E89D 00 34               FDB     52          ;DOM
0084 E89F 00 2B               FDB     43
0085 E8A1 00 35               FDB     53          ;DOM
0086 E8A3 00 2A               FDB     42
0087 E8A5 00 36               FDB     54          ;DOM
0088 E8A7 00 29               FDB     41
0089 E8A9 00 36               FDB     54          ;DOM
0090 E8AB 00 28               FDB     40
0091 E8AD 00 36               FDB     54          ;DOM
0092 E8AF 00 27               FDB     39
0093 E8B1 00 37               FDB     55          ;DOM
0094 E8B3 00 25               FDB     37
0095 E8B5 00 37               FDB     55          ;DOM
0096 E8B7 00 24               FDB     36
0097 E8B9 00 37               FDB     55          ;DOM
0098 E8BB 00 23               FDB     35
0099 E8BD 00 38               FDB     56          ;DOM
0100 E8BF 00 21               FDB     33
0101 E8C1 00 38               FDB     56          ;DOM
0102 E8C3 00 20               FDB     32
0103 E8C5 00 38               FDB     56          ;DOM
0104 E8C7 00 1F               FDB     31
0105 E8C9 00 38               FDB     56          ;DOM
0106 E8CB 00 1D               FDB     29
0107 E8CD 00 38               FDB     56          ;DOM
0108 E8CF 00 1C               FDB     28
0109                *FREQUENCY TABLE FOR 7 HZ.
0110                *
```

```
SEQ  LOC  OBJ      SOURCE

0111 E8D1 1F       FR07    FCB     31          ;NO.-PULSE PERIODS IN 60 DEG SEG
0112 E8D2 03 00            FDB     768             ;LENGTH OF PULSE PERIOD (USEC)
0113 E8D4 5D 00            FDB     23808           ;LENGTH OF 60 DEGREE SEG (USEC
0114 E8D6 07 FF            FDB     2047            ;SFMAX
0115 E8D8 00 80            FDB     128             ;SFMIN
0116                       *PULSE WIDTHS
0117 E8DA 00 37            FDB     55          ;DOM
0118 E8DC 00 36            FDB     54
0119 E8DE 00 38            FDB     56          ;DOM
0120 E8E0 00 35            FDB     53
0121 E8E2 00 39            FDB     57          ;DOM
0122 E8E4 00 34            FDB     52
0123 E8E6 00 3A            FDB     58          ;DOM
0124 E8E8 00 33            FDB     51
0125 E8EA 00 3B            FDB     59          ;DOM
0126 E8EC 00 31            FDB     49
0127 E8EE 00 3C            FDB     60          ;DOM
0128 E8F0 00 30            FDB     48
0129 E8F2 00 3C            FDB     60          ;DOM
0130 E8F4 00 2F            FDB     47
0131 E8F6 00 3D            FDB     61          ;DOM
0132 E8F8 00 2D            FDB     45
0133 E8FA 00 3E            FDB     62          ;DOM
0134 E8FC 00 2C            FDB     44
0135 E8FE 00 3E            FDB     62          ;DOM
0136 E900 00 2A            FDB     42
0137 E902 00 3E            FDB     62          ;DOM
0138 E904 00 28            FDB     40
0139 E906 00 3F            FDB     63          ;DOM
0140 E908 00 27            FDB     39
0141 E90A 00 3F            FDB     63          ;DOM
0142 E90C 00 25            FDB     37
0143 E90E 00 3F            FDB     63          ;DOM
0144 E910 00 23            FDB     35
0145 E912 00 3F            FDB     63          ;DOM
0146 E914 00 21            FDB     33
0147 E916 00 3F            FDB     63          ;DOM
0148 E918 00 1F            FDB     31
0149                       *FREQUENCY TABLE FOR 8 HZ.
0150                       *
0151 E91A 1B       FR08    FCB     27          ;NO.-PULSE PERIODS IN 60 DEG SEG
0152 E91B 03 04            FDB     772             ;LENGTH OF PULSE PERIOD (USEC)
0153 E91D 51 6C            FDB     20844           ;LENGTH OF 60 DEGREE SEG (USEC
0154 E91F 07 FF            FDB     2047            ;SFMAX
0155 E921 00 80            FDB     128             ;SFMIN
0156                       *PULSE WIDTHS
0157 E923 00 40            FDB     64          ;DOM
0158 E925 00 3E            FDB     62
0159 E927 00 41            FDB     65          ;DOM
0160 E929 00 3D            FDB     61
0161 E92B 00 42            FDB     66          ;DOM
0162 E92D 00 3B            FDB     59
0163 E92F 00 43            FDB     67          ;DOM
0164 E931 00 39            FDB     57
0165 E933 00 44            FDB     68          ;DOM
```

```
SEQ  LOC  OBJ     SOURCE

0166 E935 00 38             FDB     56
0167 E937 00 45             FDB     69          ;DOM
0168 E939 00 36             FDB     54
0169 E93B 00 46             FDB     70          ;DOM
0170 E93D 00 34             FDB     52
0171 E93F 00 47             FDB     71          ;DOM
0172 E941 00 32             FDB     50
0173 E943 00 47             FDB     71          ;DOM
0174 E945 00 30             FDB     48
0175 E947 00 48             FDB     72          ;DOM
0176 E949 00 2E             FDB     46
0177 E94B 00 48             FDB     72          ;DOM
0178 E94D 00 2B             FDB     43
0179 E94F 00 49             FDB     73          ;DOM
0180 E951 00 29             FDB     41
0181 E953 00 49             FDB     73          ;DOM
0182 E955 00 27             FDB     39
0183 E957 00 49             FDB     73          ;DOM
0184 E959 00 24             FDB     36
0185                *FREQUENCY TABLE FOR 9 HZ.
0186                *
0187 E95B 19        FRQ9    FCB     25          ;NO.-PULSE PERIODS IN 60 DEG SEG
0188 E95C 02 E5             FDB     741             ;LENGTH OF PULSE PERIOD (USEC)
0189 E95E 48 5D             FDB     18525           ;LENGTH OF 60 DEGREE SEG (USE
0190 E960 07 FF             FDB     2047            ;SFMAX
0191 E962 00 80             FDB     128             ;SFMIN
0192                *PULSE WIDTHS
0193 E964 00 45             FDB     69          ;DOM
0194 E966 00 43             FDB     67
0195 E968 00 46             FDB     70          ;DOM
0196 E96A 00 41             FDB     65
0197 E96C 00 48             FDB     72          ;DOM
0198 E96E 00 40             FDB     64
0199 E970 00 49             FDB     73          ;DOM
0200 E972 00 3E             FDB     62
0201 E974 00 4A             FDB     74          ;DOM
0202 E976 00 3B             FDB     59
0203 E978 00 4B             FDB     75          ;DOM
0204 E97A 00 39             FDB     57
0205 E97C 00 4C             FDB     76          ;DOM
0206 E97E 00 37             FDB     55
0207 E980 00 4D             FDB     77          ;DOM
0208 E982 00 35             FDB     53
0209 E984 00 4E             FDB     78          ;DOM
0210 E986 00 32             FDB     50
0211 E988 00 4E             FDB     78          ;DOM
0212 E98A 00 2F             FDB     47
0213 E98C 00 4E             FDB     78          ;DOM
0214 E98E 00 2D             FDB     45
0215 E990 00 4F             FDB     79          ;DOM
0216 E992 00 2A             FDB     42
0217 E994 00 4F             FDB     79          ;DOM
0218 E996 00 27             FDB     39
0219                *FREQUENCY TABLE FOR 10 HZ.
0220                *
```

```
SEQ  LOC  OBJ      SOURCE

0221 E998 15       FR010   FCB    21          ;NO.-PULSE PERIODS IN 60 DEG SEG
0222 E999 03 1A            FDB    794           ;LENGTH OF PULSE PERIOD (USEC)
0223 E99B 41 22            FDB    16674         ;LENGTH OF 60 DEGREE SEG (USEC)
0224 E99D 07 FF            FDB    2047          ;SFMAX
0225 E99F 00 80            FDB    128           ;SFMIN
0226                       *PULSE WIDTHS
0227 E9A1 00 52            FDB    82          ;DOM
0228 E9A3 00 50            FDB    80
0229 E9A5 00 54            FDB    84          ;DOM
0230 E9A7 00 4D            FDB    77
0231 E9A9 00 56            FDB    86          ;DOM
0232 E9AB 00 4B            FDB    75
0233 E9AD 00 58            FDB    88          ;DOM
0234 E9AF 00 48            FDB    72
0235 E9B1 00 5A            FDB    90          ;DOM
0236 E9B3 00 45            FDB    69
0237 E9B5 00 5B            FDB    91          ;DOM
0238 E9B7 00 41            FDB    65
0239 E9B9 00 5C            FDB    92          ;DOM
0240 E9BB 00 3E            FDB    62
0241 E9BD 00 5D            FDB    93          ;DOM
0242 E9BF 00 3A            FDB    58
0243 E9C1 00 5D            FDB    93          ;DOM
0244 E9C3 00 37            FDB    55
0245 E9C5 00 5E            FDB    94          ;DOM
0246 E9C7 00 33            FDB    51
0247 E9C9 00 5E            FDB    94          ;DOM
0248 E9CB 00 2F            FDB    47
0249                       *FREQUENCY TABLE FOR 11 HZ.
0250                       *
0251 E9CD 13       FR011   FCB    19          ;NO.-PULSE PERIODS IN 60 DEG SEG
0252 E9CE 03 1D            FDB    797           ;LENGTH OF PULSE PERIOD (USEC)
0253 E9D0 3B 27            FDB    15143         ;LENGTH OF 60 DEGREE SEG (USEC)
0254 E9D2 07 6A            FDB    1898          ;SFMAX
0255 E9D4 00 80            FDB    128           ;SFMIN
0256                       *PULSE WIDTHS
0257 E9D6 00 5B            FDB    91          ;DOM
0258 E9D8 00 58            FDB    88
0259 E9DA 00 5E            FDB    94          ;DOM
0260 E9DC 00 55            FDB    85
0261 E9DE 00 60            FDB    96          ;DOM
0262 E9E0 00 52            FDB    82
0263 E9E2 00 62            FDB    98          ;DOM
0264 E9E4 00 4E            FDB    78
0265 E9E6 00 64            FDB    100         ;DOM
0266 E9E8 00 4A            FDB    74
0267 E9EA 00 65            FDB    101         ;DOM
0268 E9EC 00 46            FDB    70
0269 E9EE 00 66            FDB    102         ;DOM
0270 E9F0 00 42            FDB    66
0271 E9F2 00 67            FDB    103         ;DOM
0272 E9F4 00 3D            FDB    61
0273 E9F6 00 68            FDB    104         ;DOM
0274 E9F8 00 39            FDB    57
0275 E9FA 00 68            FDB    104         ;DOM
```

```
SEQ  LOC  OBJ      SOURCE

0276 E9FC 00 34            FDB    52
0277               *FREQUENCY TABLE FOR 12 HZ.
0278               *
0279 E9FE 13       FR012   FCB    19          ;NO.-PULSE PERIODS IN 60 DEG SEG
0280 E9FF 02 DB            FDB    731            ;LENGTH OF PULSE PERIOD (USEC)
0281 EA01 36 41            FDB    13889          ;LENGTH OF 60 DEGREE SEG (USEC
0282 EA03 06 C7            FDB    1735           ;SFMAX
0283 EA05 00 80            FDB    128            ;SFMIN
0284               *PULSE WIDTHS
0285 EA07 00 5B            FDB    91          ;DOM
0286 EA09 00 58            FDB    88
0287 EA0B 00 5E            FDB    94          ;DOM
0288 EA0D 00 55            FDB    85
0289 EA0F 00 60            FDB    96          ;DOM
0290 EA11 00 52            FDB    82
0291 EA13 00 62            FDB    98          ;DOM
0292 EA15 00 4E            FDB    78
0293 EA17 00 64            FDB    100         ;DOM
0294 EA19 00 4A            FDB    74
0295 EA1B 00 65            FDB    101         ;DOM
0296 EA1D 00 46            FDB    70
0297 EA1F 00 66            FDB    102         ;DOM
0298 EA21 00 42            FDB    66
0299 EA23 00 67            FDB    103         ;DOM
0300 EA25 00 3D            FDB    61
0301 EA27 00 68            FDB    104         ;DOM
0302 EA29 00 39            FDB    57
0303 EA2B 00 68            FDB    104         ;DOM
0304 EA2D 00 34            FDB    52
0305               *FREQUENCY TABLE FOR 13 HZ.
0306               *
0307 EA2F 11       FR013   FCB    17          ;NO.-PULSE PERIODS IN 60 DEG SEG
0308 EA30 02 F2            FDB    754            ;LENGTH OF PULSE PERIOD (USEC)
0309 EA32 32 12            FDB    12818          ;LENGTH OF 60 DEGREE SEG (USE
0310 EA34 06 47            FDB    1607           ;SFMAX
0311 EA36 00 80            FDB    128            ;SFMIN
0312               *PULSE WIDTHS
0313 EA38 00 66            FDB    102         ;DOM
0314 EA3A 00 63            FDB    99
0315 EA3C 00 69            FDB    105         ;DOM
0316 EA3E 00 5F            FDB    95
0317 EA40 00 6C            FDB    108         ;DOM
0318 EA42 00 5A            FDB    90
0319 EA44 00 6F            FDB    111         ;DOM
0320 EA46 00 56            FDB    86
0321 EA48 00 70            FDB    112         ;DOM
0322 EA4A 00 51            FDB    81
0323 EA4C 00 72            FDB    114         ;DOM
0324 EA4E 00 4B            FDB    75
0325 EA50 00 73            FDB    115         ;DOM
0326 EA52 00 46            FDB    70
0327 EA54 00 74            FDB    116         ;DOM
0328 EA56 00 40            FDB    64
0329 EA58 00 74            FDB    116         ;DOM
0330 EA5A 00 3A            FDB    58
```

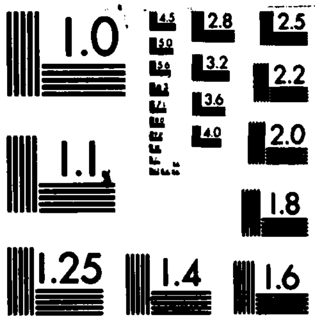
```
SEQ  LOC  OBJ      SOURCE

                   *FREQUENCY TABLE FOR 14 HZ.
                   *
     EA5C 0F       FRQ14   FCB     15          ;NO.-PULSE PERIODS IN 60 DEG SEG
     EA5D 03 1A            FDB     794             ;LENGTH OF PULSE PERIOD (USEC)
     EA5F 2E 86            FDB     11910           ;LENGTH OF 60 DEGREE SEG (USEC
     EA61 05 D1            FDB     1489            ;SFMAX
     EA63 00 80            FDB     128             ;SFMIN
                   *PULSE WIDTHS
     EA65 00 74            FDB     116         ;DOM
     EA67 00 6F            FDB     111
     EA69 00 78            FDB     120         ;DOM
     EA6B 00 6A            FDB     106
     EA6D 00 7C            FDB     124         ;DOM
     EA6F 00 65            FDB     101
     EA71 00 7E            FDB     126         ;DOM
     EA73 00 5E            FDB     94
     EA75 00 81            FDB     129         ;DOM
     EA77 00 58            FDB     88
     EA79 00 82            FDB     130         ;DOM
     EA7B 00 51            FDB     81
     EA7D 00 83            FDB     131         ;DOM
     EA7F 00 49            FDB     73
     EA81 00 84            FDB     132         ;DOM
     EA83 00 42            FDB     66
                   *FREQUENCY TABLE FOR 15 HZ.
                   *
     EA85 0F       FRQ15   FCB     15          ;NO.-PULSE PERIODS IN 60 DEG SEG
     EA86 02 E5            FDB     741             ;LENGTH OF PULSE PERIOD (USEC)
     EA88 2B 6B            FDB     11115           ;LENGTH OF 60 DEGREE SEG (USEC
     EA8A 05 6B            FDB     1387            ;SFMAX
     EA8C 00 80            FDB     128             ;SFMIN
                   *PULSE WIDTHS
     EA8E 00 74            FDB     116         ;DOM
     EA90 00 6F            FDB     111
     EA92 00 78            FDB     120         ;DOM
     EA94 00 6A            FDB     106
     EA96 00 7C            FDB     124         ;DOM
     EA98 00 65            FDB     101
     EA9A 00 7E            FDB     126         ;DOM
     EA9C 00 5E            FDB     94
     EA9E 00 81            FDB     129         ;DOM
     EAA0 00 58            FDB     88
     EAA2 00 82            FDB     130         ;DOM
     EAA4 00 51            FDB     81
     EAA6 00 83            FDB     131         ;DOM
     EAA8 00 49            FDB     73
     EAAA 00 84            FDB     132         ;DOM
     EAAC 00 42            FDB     66
                   *FREQUENCY TABLE FOR 16 HZ.
                   *
     EAAE 0F       FRQ16   FCB     15          ;NO.-PULSE PERIODS IN 60 DEG SEG
     EAAF 02 B6            FDB     694             ;LENGTH OF PULSE PERIOD (USEC
     EAB1 28 AA            FDB     10410           ;LENGTH OF 60 DEGREE SEG (USE
     EAB3 05 10            FDB     1296            ;SFMAX
     EAB5 00 80            FDB     128             ;SFMIN
```

```
SEQ  LOC  OBJ       SOURCE

0386                *PULSE WIDTHS
0387 EAB7 00 74              FDB     116          ;DOM
0388 EAB9 00 6F              FDB     111
0389 EABB 00 78              FDB     120          ;DOM
0390 EABD 00 6A              FDB     106
0391 EABF 00 7C              FDB     124          ;DOM
0392 EAC1 00 65              FDB     101
0393 EAC3 00 7E              FDB     126          ;DOM
0394 EAC5 00 5E              FDB     94
0395 EAC7 00 81              FDB     129          ;DOM
0396 EAC9 00 58              FDB     88
0397 EACB 00 82              FDB     130          ;DOM
0398 EACD 00 51              FDB     81
0399 EACF 00 83              FDB     131          ;DOM
0400 EAD1 00 49              FDB     73
0401 EAD3 00 84              FDB     132          ;DOM
0402 EAD5 00 42              FDB     66
0403                *FREQUENCY TABLE FOR 17 HZ.
0404                *
0405 EAD7 0D        FRQ17    FCB     13           ;NO. -PULSE PERIODS IN 60 DEG SEG
0406 EAD8 02 F2              FDB     754             ;LENGTH OF PULSE PERIOD (USEC
0407 EADA 26 4A              FDB     9802            ;LENGTH OF 60 DEGREE SEG (USEC)
0408 EADC 04 CA              FDB     1226            ;SFMAX
0409 EADE 00 80              FDB     128             ;SFMIN
0410                *PULSE WIDTHS
0411 EAE0 00 86              FDB     134          ;DOM
0412 EAE2 00 80              FDB     128
0413 EAE4 00 8C              FDB     140          ;DOM
0414 EAE6 00 79              FDB     121
0415 EAE8 00 90              FDB     144          ;DOM
0416 EAEA 00 72              FDB     114
0417 EAEC 00 93              FDB     147          ;DOM
0418 EAEE 00 69              FDB     105
0419 EAF0 00 96              FDB     150          ;DOM
0420 EAF2 00 60              FDB     96
0421 EAF4 00 97              FDB     151          ;DOM
0422 EAF6 00 56              FDB     86
0423 EAF8 00 98              FDB     152          ;DOM
0424 EAFA 00 4C              FDB     76
0425                *FREQUENCY TABLE FOR 18 HZ.
0426                *
0427 EAFC 0D        FRQ18    FCB     13           ;NO. -PULSE PERIODS IN 60 DEG SEG
0428 EAFD 02 C8              FDB     712             ;LENGTH OF PULSE PERIOD (USEC)
0429 EAFF 24 28              FDB     9256            ;LENGTH OF 60 DEGREE SEG (USE
0430 EB01 04 83              FDB     1155            ;SFMAX
0431 EB03 00 80              FDB     128             ;SFMIN
0432                *PULSE WIDTHS
0433 EB05 00 86              FDB     134          ;DOM
0434 EB07 00 80              FDB     128
0435 EB09 00 8C              FDB     140          ;DOM
0436 EB0B 00 79              FDB     121
0437 EB0D 00 90              FDB     144          ;DOM
0438 EB0F 00 72              FDB     114
0439 EB11 00 93              FDB     147          ;DOM
0440 EB13 00 69              FDB     105
```

```
SEQ  LOC  OBJ        SOURCE

0441 EB15 00 96                FDB     150           ;DOM
0442 EB17 00 60                FDB     96
0443 EB19 00 97                FDB     151           ;DOM
0444 EB1B 00 56                FDB     86
0445 EB1D 00 98                FDB     152           ;DOM
0446 EB1F 00 4C                FDB     76
0447                 *FREQUENCY TABLE FOR 19 HZ.
0448                 *
0449 EB21 0B         FR019     FCB     11            ;NO. -PULSE PERIODS IN 60 DEG SEG
0450 EB22 03 1D                FDB     797              ;LENGTH OF PULSE PERIOD (USEC)
0451 EB24 22 3F                FDB     8767             ;LENGTH OF 60 DEGREE SEG (USEC
0452 EB26 04 4F                FDB     1103             ;SFMAX
0453 EB28 00 80                FDB     128              ;SFMIN
0454                 *PULSE WIDTHS
0455 EB2A 00 9F                FDB     159           ;DOM
0456 EB2C 00 97                FDB     151
0457 EB2E 00 A7                FDB     167           ;DOM
0458 EB30 00 8D                FDB     141
0459 EB32 00 AC                FDB     172           ;DOM
0460 EB34 00 82                FDB     130
0461 EB36 00 B0                FDB     176           ;DOM
0462 EB38 00 75                FDB     117
0463 EB3A 00 B3                FDB     179           ;DOM
0464 EB3C 00 68                FDB     104
0465 EB3E 00 B3                FDB     179           ;DOM
0466 EB40 00 5A                FDB     90
0467                 *FREQUENCY TABLE FOR 20 HZ.
0468                 *
0469 EB42 0B         FR020     FCB     11            ;NO. -PULSE PERIODS IN 60 DEG SEG
0470 EB43 02 F6                FDB     758              ;LENGTH OF PULSE PERIOD (USEC)
0471 EB45 20 92                FDB     8338             ;LENGTH OF 60 DEGREE SEG (USEC
0472 EB47 04 17                FDB     1047             ;SFMAX
0473 EB49 00 80                FDB     128              ;SFMIN
0474                 *PULSE WIDTHS
0475 EB4B 00 9F                FDB     159           ;DOM
0476 EB4D 00 97                FDB     151
0477 EB4F 00 A7                FDB     167           ;DOM
0478 EB51 00 8D                FDB     141
0479 EB53 00 AC                FDB     172           ;DOM
0480 EB55 00 82                FDB     130
0481 EB57 00 B0                FDB     176           ;DOM
0482 EB59 00 75                FDB     117
0483 EB5B 00 B3                FDB     179           ;DOM
0484 EB5D 00 68                FDB     104
0485 EB5F 00 B3                FDB     179           ;DOM
0486 EB61 00 5A                FDB     90
0487                 *FREQUENCY TABLE FOR 21 HZ.
0488                 *
0489 EB63 0B         FR021     FCB     11            ;NO. -PULSE PERIODS IN 60 DEG SEG
0490 EB64 02 D2                FDB     722              ;LENGTH OF PULSE PERIOD (USEC
0491 EB66 1F 06                FDB     7942             ;LENGTH OF 60 DEGREE SEG (USE
0492 EB68 03 E3                FDB     995              ;SFMAX
0493 EB6A 00 80                FDB     128              ;SFMIN
0494                 *PULSE WIDTHS
0495 EB6C 00 9F                FDB     159           ;DOM
```

AD-A140 299 ADVANCED MOTOR-CONTROLLER DEVELOPMENT(U) WESTINGHOUSE OCEANIC DIV ANNAPOLIS MD L E LESSTER ET AL. 22 JUN 83 DTNSRDC-PASD-CR-1-83 N00167-82-C-0122 3/4

UNCLASSIFIED F/G 9/3 NL



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

```
SEQ  LOC  OBJ      SOURCE

0496 EB6E 00 97             FDB     151
0497 EB70 00 A7             FDB     167          ;DOM
0498 EB72 00 8D             FDB     141
0499 EB74 00 AC             FDB     172          ;DOM
0500 EB76 00 82             FDB     130
0501 EB78 00 B0             FDB     176          ;DOM
0502 EB7A 00 75             FDB     117
0503 EB7C 00 B3             FDB     179          ;DOM
0504 EB7E 00 68             FDB     104
0505 EB80 00 B3             FDB     179          ;DOM
0506 EB82 00 5A             FDB     90
0507               *FREQUENCY TABLE FOR 22 HZ.
0508               *
0509 EB84 0B       FRQ22    FCB     11           ;NO. -PULSE PERIODS IN 60 DEG SEG
0510 EB85 02 B1             FDB     689              ;LENGTH OF PULSE PERIOD (USEC)
0511 EB87 1D 9B             FDB     7579             ;LENGTH OF 60 DEGREE SEG (USEC)
0512 EB89 03 B4             FDB     948              ;SFMAX
0513 EB8B 00 80             FDB     128              ;SFMIN
0514               *PULSE WIDTHS
0515 EB8D 00 9F             FDB     159          ;DOM
0516 EB8F 00 97             FDB     151
0517 EB91 00 A7             FDB     167          ;DOM
0518 EB93 00 8D             FDB     141
0519 EB95 00 AC             FDB     172          ;DOM
0520 EB97 00 82             FDB     130
0521 EB99 00 B0             FDB     176          ;DOM
0522 EB9B 00 75             FDB     117
0523 EB9D 00 B3             FDB     179          ;DOM
0524 EB9F 00 68             FDB     104
0525 EBA1 00 B3             FDB     179          ;DOM
0526 EBA3 00 5A             FDB     90
0527               *FREQUENCY TABLE FOR 23 HZ.
0528               *
0529 EBA5 0B       FRQ23    FCB     11           ;NO. -PULSE PERIODS IN 60 DEG SEG
0530 EBA6 02 93             FDB     659              ;LENGTH OF PULSE PERIOD (USEC)
0531 EBA8 1C 51             FDB     7249             ;LENGTH OF 60 DEGREE SEG (USEC)
0532 EBAA 03 89             FDB     905              ;SFMAX
0533 EBAC 00 80             FDB     128              ;SFMIN
0534               *PULSE WIDTHS
0535 EBAE 00 9F             FDB     159          ;DOM
0536 EBB0 00 97             FDB     151
0537 EBB2 00 A7             FDB     167          ;DOM
0538 EBB4 00 8D             FDB     141
0539 EBB6 00 AC             FDB     172          ;DOM
0540 EBB8 00 82             FDB     130
0541 EBBA 00 B0             FDB     176          ;DOM
0542 EBBC 00 75             FDB     117
0543 EBBE 00 B3             FDB     179          ;DOM
0544 EBC0 00 68             FDB     104
0545 EBC2 00 B3             FDB     179          ;DOM
0546 EBC4 00 5A             FDB     90
0547               *FREQUENCY TABLE FOR 24 HZ.
0548               *
0549 EBC6 0B       FRQ24    FCB     11           ;NO. -PULSE PERIODS IN 60 DEG SEG
0550 EBC7 02 77             FDB     631              ;LENGTH OF PULSE PERIOD (USEC
```

```
SEQ  LOC  OBJ      SOURCE

0551 EBC9 1B 1D              FDB     6941        ;LENGTH OF 60 DEGREE SEG (USEC
0552 EBCB 03 61              FDB     865         ;SFM
0553 EBCD 00 80              FDB     128         ;SFMIN
0554                *PULSE WIDTHS
0555 EBCF 00 A0              FDB     160      ;DOM
0556 EBD1 00 97              FDB     151
0557 EBD3 00 A7              FDB     167      ;DOM
0558 EBD5 00 8D              FDB     141
0559 EBD7 00 AC              FDB     172      ;DOM
0560 EBD9 00 82              FDB     130
0561 EBDB 00 B0              FDB     176      ;DOM
0562 EBDD 00 75              FDB     117
0563 EBDF 00 B3              FDB     179      ;DOM
0564 EBE1 00 68              FDB     104
0565 EBE3 00 B3              FDB     179      ;DOM
0566 EBE5 00 5A              FDB     90
0567                *FREQUENCY TABLE FOR 25 HZ.
0568                *
0569 EBE7 09        FRQ25    FCB     9        ;NO.-PULSE PERIODS IN 60 DEG SEG
0570 EBE8 02 E5              FDB     741         ;LENGTH OF PULSE PERIOD (USEC)
0571 EBEA 1A 0D              FDB     6669        ;LENGTH OF 60 DEGREE SEG (USEC
0572 EBEC 03 44              FDB     836         ;SFMAX
0573 EBEE 00 80              FDB     128         ;SFMIN
0574                *PULSE WIDTHS
0575 EBF0 00 C4              FDB     196      ;DOM
0576 EBF2 00 B7              FDB     183
0577 EBF4 00 CE              FDB     206      ;DOM
0578 EBF6 00 A8              FDB     168
0579 EBF8 00 D6              FDB     214      ;DOM
0580 EBFA 00 96              FDB     150
0581 EBFC 00 DA              FDB     218      ;DOM
0582 EBFE 00 83              FDB     131
0583 EC00 00 DB              FDB     219      ;DOM
0584 EC02 00 6E              FDB     110
0585                *FREQUENCY TABLE FOR 26 HZ.
0586                *
0587 EC04 09        FRQ26    FCB     9        ;NO.-PULSE PERIODS IN 60 DEG SEG
0588 EC05 02 C8              FDB     712         ;LENGTH OF PULSE PERIOD (USEC
0589 EC07 19 08              FDB     6408        ;LENGTH OF 60 DEGREE SEG (USE
0590 EC09 03 22              FDB     802         ;SFMAX
0591 EC0B 00 80              FDB     128         ;SFMIN
0592                *PULSE WIDTHS
0593 EC0D 00 C4              FDB     196      ;DOM
0594 EC0F 00 B7              FDB     183
0595 EC11 00 CE              FDB     206      ;DOM
0596 EC13 00 A8              FDB     168
0597 EC15 00 D6              FDB     214      ;DOM
0598 EC17 00 96              FDB     150
0599 EC19 00 DA              FDB     218      ;DOM
0600 EC1B 00 83              FDB     131
0601 EC1D 00 DB              FDB     219      ;DOM
0602 EC1F 00 6E              FDB     110
0603                *FREQUENCY TABLE FOR 27 HZ.
0604                *
0605 EC21 09        FRQ27    FCB     9        ;NO.-PULSE PERIODS IN 60 DEG SEG
```

```
LOC  OBJ       SOURCE

EC22 02 AE               FDB     686            ;LENGTH OF PULSE PERIOD (USEC
EC24 18 1E               FDB     6174           ;LENGTH OF 60 DEGREE SEG (USEC
EC26 03 04               FDB     772            ;SFMAX
EC28 00 80               FDB     128            ;SFMIN
               *PULSE WIDTHS
EC2A 00 C4               FDB     196       ;DOM
EC2C 00 B7               FDB     183
EC2E 00 CE               FDB     206       ;DOM
EC30 00 A8               FDB     168
EC32 00 D6               FDB     214       ;DOM
EC34 00 96               FDB     150
EC36 00 DA               FDB     218       ;DOM
EC38 00 83               FDB     131
EC3A 00 DB               FDB     219       ;DOM
EC3C 00 6E               FDB     110
               *FREQUENCY TABLE FOR 28 HZ.
               *
EC3E 09        FRQ28     FCB     9         ;NO.-PULSE PERIODS IN 60 DEG SEG
EC3F 02 95               FDB     661            ;LENGTH OF PULSE PERIOD (USEC)
EC41 17 3D               FDB     5949           ;LENGTH OF 60 DEGREE SEG (USE
EC43 02 E6               FDB     742            ;SFMAX
EC45 00 80               FDB     128            ;SFMIN
               *PULSE WIDTHS
EC47 00 C4               FDB     196       ;DOM
EC49 00 B7               FDB     183
EC4B 00 CE               FDB     206       ;DOM
EC4D 00 A8               FDB     168
EC4F 00 D6               FDB     214       ;DOM
EC51 00 96               FDB     150
EC53 00 DA               FDB     218       ;DOM
EC55 00 83               FDB     131
EC57 00 DB               FDB     219       ;DOM
EC59 00 6E               FDB     110
               *FREQUENCY TABLE FOR 29 HZ.
               *
EC5B 09        FRQ29     FCB     9         ;NO.-PULSE PERIODS IN 60 DEG SEG
EC5C 02 7F               FDB     639            ;LENGTH OF PULSE PERIOD (USEC
EC5E 16 77               FDB     5751           ;LENGTH OF 60 DEGREE SEG (USE
EC60 02 CD               FDB     717            ;SFMAX
EC62 00 80               FDB     128            ;SFMIN
               *PULSE WIDTHS
EC64 00 C4               FDB     196       ;DOM
EC66 00 B7               FDB     183
EC68 00 CE               FDB     206       ;DOM
EC6A 00 A8               FDB     168
EC6C 00 D6               FDB     214       ;DOM
EC6E 00 96               FDB     150
EC70 00 DA               FDB     218       ;DOM
EC72 00 83               FDB     131
EC74 00 DB               FDB     219       ;DOM
EC76 00 6E               FDB     110
               *FREQUENCY TABLE FOR 30 HZ.
               *
EC78 07        FRQ30     FCB     7         ;NO.-PULSE PERIODS IN 60 DEG SEG
EC79 03 1A               FDB     794            ;LENGTH OF PULSE PERIOD (USE
```

```
SEQ  LOC  OBJ      SOURCE

0661 EC7B 15 B6              FDB     5558              ;LENGTH OF 60 DEGREE SEG (USEC
0662 EC7D 02 B9              FDB     697               ;SFMAX
0663 EC7F 00 80              FDB     128               ;SFMIN
0664                 *PULSE WIDTHS
0665 EC81 00 FE              FDB     254           ;DOM
0666 EC83 00 E9              FDB     233
0667 EC85 01 0E              FDB     270           ;DOM
0668 EC87 00 CF              FDB     207
0669 EC89 01 17              FDB     279           ;DOM
0670 EC8B 00 B0              FDB     176
0671 EC8D 01 1A              FDB     282           ;DOM
0672 EC8F 00 8D              FDB     141
0673                 *FREQUENCY TABLE FOR 31 HZ.
0674                 *
0675 EC91 07         FRQ31   FCB     7             ;NO.-PULSE PERIODS IN 60 DEG SEG
0676 EC92 03 00              FDB     768 .             ;LENGTH OF PULSE PERIOD (USEC)
0677 EC94 15 00              FDB     5376              ;LENGTH OF 60 DEGREE SEG (USEC
0678 EC96 02 A2              FDB     674               ;SFMAX
0679 EC98 00 80              FDB     128               ;SFMIN
0680                 *PULSE WIDTHS
0681 EC9A 00 FE              FDB     254           ;DOM
0682 EC9C 00 E9              FDB     233
0683 EC9E 01 0E              FDB     270           ;DOM
0684 ECA0 00 CF              FDB     207
0685 ECA2 01 17              FDB     279           ;DOM
0686 ECA4 00 B0              FDB     176
0687 ECA6 01 1A              FDB     282           ;DOM
0688 ECA8 00 8D              FDB     141
0689                 *FREQUENCY TABLE FOR 32 HZ.
0690                 *
0691 ECAA 07         FRQ32   FCB     7             ;NO.-PULSE PERIODS IN 60 DEG SEG
0692 ECAB 02 E8              FDB     744               ;LENGTH OF PULSE PERIOD (USEC)
0693 ECAD 14 58              FDB     5208              ;LENGTH OF 60 DEGREE SEG (USEC
0694 ECAF 02 8C              FDB     652               ;SFMAX
0695 ECB1 00 80              FDB     128               ;SFMIN
0696                 *PULSE WIDTHS
0697 ECB3 00 FE              FDB     254           ;DOM
0698 ECB5 00 E9              FDB     233
0699 ECB7 01 0E              FDB     270           ;DOM
0700 ECB9 00 CF              FDB     207
0701 ECBB 01 17              FDB     279           ;DOM
0702 ECBD 00 B0              FDB     176
0703 ECBF 01 1A              FDB     282           ;DOM
0704 ECC1 00 8D              FDB     141
0705                 *FREQUENCY TABLE FOR 33 HZ.
0706                 *
0707 ECC3 07         FRQ33   FCB     7             ;NO.-PULSE PERIODS IN 60 DEG SEG
0708 ECC4 02 D2              FDB     722               ;LENGTH OF PULSE PERIOD (USEC
0709 ECC6 13 BE              FDB     5054              ;LENGTH OF 60 DEGREE SEG (USE
0710 ECC8 02 78              FDB     632               ;SFMAX
0711 ECCA 00 80              FDB     128               ;SFMIN
0712                 *PULSE WIDTHS
0713 ECCC 00 FE              FDB     254           ;DOM
0714 ECCE 00 E9              FDB     233
0715 ECD0 01 0E              FDB     270           ;DOM
```

```
SEQ  LOC  OBJ      SOURCE

0716 ECD2 00 CF            FDB     207
0717 ECD4 01 17            FDB     279         ;DOM
0718 ECD6 00 B0            FDB     176
0719 ECD8 01 1A            FDB     282         ;DOM
0720 ECDA 00 8D            FDB     141
0721                 *FREQUENCY TABLE FOR 34 HZ.
0722                 *
0723 ECDC 07         FRQ34  FCB     7           ;NO.-PULSE PERIODS IN 60 DEG SEG
0724 ECDD 02 BC            FDB     700            ;LENGTH OF PULSE PERIOD (USEC)
0725 ECDF 13 24            FDB     4900           ;LENGTH OF 60 DEGREE SEG (USE
0726 ECE1 02 64            FDB     612            ;SFMAX
0727 ECE3 00 80            FDB     128            ;SFMIN
0728                 *PULSE WIDTHS
0729 ECE5 00 FE            FDB     254         ;DOM
0730 ECE7 00 E9            FDB     233
0731 ECE9 01 0E            FDB     270         ;DOM
0732 ECEB 00 CF            FDB     207
0733 ECED 01 17            FDB     279         ;DOM
0734 ECEF 00 B0            FDB     176
0735 ECF1 01 1A            FDB     282         ;DOM
0736 ECF3 00 8D            FDB     141
0737                 *FREQUENCY TABLE FOR 35 HZ.
0738                 *
0739 ECF5 07         FRQ35  FCB     7           ;NO.-PULSE PERIODS IN 60 DEG SEG
0740 ECF6 02 A8            FDB     680            ;LENGTH OF PULSE PERIOD (USEC
0741 ECF8 12 98            FDB     4760           ;LENGTH OF 60 DEGREE SEG (USE
0742 ECFA 02 52            FDB     594            ;SFMAX
0743 ECFC 00 80            FDB     128            ;SFMIN
0744                 *PULSE WIDTHS
0745 ECFE 00 FE            FDB     254         ;DOM
0746 ED00 00 E9            FDB     233
0747 ED02 01 0E            FDB     270         ;DOM
0748 ED04 00 CF            FDB     207
0749 ED06 01 17            FDB     279         ;DOM
0750 ED08 00 B0            FDB     176
0751 ED0A 01 1A            FDB     282         ;DOM
0752 ED0C 00 8D            FDB     141
0753                 *FREQUENCY TABLE FOR 36 HZ.
0754                 *
0755 ED0E 07         FRQ36  FCB     7           ;NO.-PULSE PERIODS IN 60 DEG SEG
0756 ED0F 02 95            FDB     661            ;LENGTH OF PULSE PERIOD (USEC
0757 ED11 12 13            FDB     4627           ;LENGTH OF 60 DEGREE SEG (US
0758 ED13 02 40            FDB     576            ;SFMAX
0759 ED15 00 80            FDB     128            ;SFMIN
0760                 *PULSE WIDTHS
0761 ED17 00 FE            FDB     254         ;DOM
0762 ED19 00 E9            FDB     233
0763 ED1B 01 0E            FDB     270         ;DOM
0764 ED1D 00 CF            FDB     207
0765 ED1F 01 17            FDB     279         ;DOM
0766 ED21 00 B0            FDB     176
0767 ED23 01 1A            FDB     282         ;DOM
0768 ED25 00 8D            FDB     141
0769                 *FREQUENCY TABLE FOR 37 HZ.
0770                 *
```

```
SEQ  LOC  OBJ        SOURCE

0771 ED27 07         FRQ37   FCB     7          ;NO.-PULSE PERIODS IN 60 DEG SEG
0772 ED28 02 84              FDB     644            ;LENGTH OF PULSE PERIOD (USEC)
0773 ED2A 11 9C              FDB     4508           ;LENGTH OF 60 DEGREE SEG (USEC
0774 ED2C 02 31              FDB     561            ;SFMAX
0775 ED2E 00 80              FDB     128            ;SFMIN
0776                 *PULSE WIDTHS
0777 ED30 00 FE              FDB     254        ;DOM
0778 ED32 00 E9              FDB     233
0779 ED34 01 0E              FDB     270        ;DOM
0780 ED36 00 CF              FDB     207
0781 ED38 01 17              FDB     279        ;DOM
0782 ED3A 00 B0              FDB     176
0783 ED3C 01 1A              FDB     282        ;DOM
0784 ED3E 00 8D              FDB     141
0785                 *FREQUENCY TABLE FOR 38 HZ.
0786                 *
0787 ED40 07         FRQ38   FCB     7          ;NO.-PULSE PERIODS IN 60 DEG SEG
0788 ED41 02 73              FDB     627            ;LENGTH OF PULSE PERIOD (USEC)
0789 ED43 11 25              FDB     4389           ;LENGTH OF 60 DEGREE SEG (USEC)
0790 ED45 02 22              FDB     546            ;SFMAX
0791 ED47 00 80              FDB     128            ;SFMIN
0792                 *PULSE WIDTHS
0793 ED49 00 FE              FDB     254        ;DOM
0794 ED4B 00 E9              FDB     233
0795 ED4D 01 0E              FDB     270        ;DOM
0796 ED4F 00 CF              FDB     207
0797 ED51 01 17              FDB     279        ;DOM
0798 ED53 00 B0              FDB     176
0799 ED55 01 1A              FDB     282        ;DOM
0800 ED57 00 8D              FDB     141
0801                 *FREQUENCY TABLE FOR 39 HZ.
0802                 *
0803 ED59 07         FRQ39   FCB     7          ;NO.-PULSE PERIODS IN 60 DEG SEG
0804 ED5A 02 63              FDB     611            ;LENGTH OF PULSE PERIOD (USEC)
0805 ED5C 10 B5              FDB     4277           ;LENGTH OF 60 DEGREE SEG (USEC)
0806 ED5E 02 13              FDB     531            ;SFMAX
0807 ED60 00 80              FDB     128            ;SFMIN
0808                 *PULSE WIDTHS
0809 ED62 00 FE              FDB     254        ;DOM
0810 ED64 00 E9              FDB     233
0811 ED66 01 0E              FDB     270        ;DOM
0812 ED68 00 CF              FDB     207
0813 ED6A 01 17              FDB     279        ;DOM
0814 ED6C 00 B0              FDB     176
0815 ED6E 01 1A              FDB     282        ;DOM
0816 ED70 00 8D              FDB     141
0817                 *FREQUENCY TABLE FOR 40 HZ.
0818                 *
0819 ED72 07         FRQ40   FCB     7          ;NO.-PULSE PERIODS IN 60 DEG SEG
0820 ED73 02 53              FDB     595            ;LENGTH OF PULSE PERIOD (USEC)
0821 ED75 10 45              FDB     4165           ;LENGTH OF 60 DEGREE SEG (USEC
0822 ED77 02 05              FDB     517            ;SFMAX
0823 ED79 00 80              FDB     128            ;SFMIN
0824                 *PULSE WIDTHS
0825 ED7B 00 FE              FDB     254        ;DOM
```

```
LOC  OBJ      SOURCE

ED7D 00 E9              FDB     233
ED7F 01 0E              FDB     270         ;DOM
ED81 00 CF              FDB     207
ED83 01 17              FDB     279         ;DOM
ED85 00 B0              FDB     176
ED87 01 1A              FDB     282         ;DOM
ED89 00 8D              FDB     141
              *FREQUENCY TABLE FOR 41 HZ.
              *
ED8B 07       FRQ41     FCB     7           ;NO.-PULSE PERIODS IN 60 DEG SEG
ED8C 02 45              FDB     581             ;LENGTH OF PULSE PERIOD (USE(
ED8E 0F E3              FDB     4067            ;LENGTH OF 60 DEGREE SEG (USEC
ED90 01 F8              FDB     504             ;SFMAX
ED92 00 80              FDB     128             ;SFMIN
              *PULSE WIDTHS
ED94 00 FE              FDB     254         ;DOM
ED96 00 E9              FDB     233
ED98 01 0E              FDB     270         ;DOM
ED9A 00 CF              FDB     207
ED9C 01 17              FDB     279         ;DOM
ED9E 00 B0              FDB     176
EDA0 01 1A              FDB     282         ;DOM
EDA2 00 8D              FDB     141
              *FREQUENCY TABLE FOR 42 HZ.
              *
EDA4 05       FRQ42     FCB     5           ;NO.-PULSE PERIODS IN 60 DEG SEG
EDA5 03 1A              FDB     794             ;LENGTH OF PULSE PERIOD (USE(
EDA7 0F 82              FDB     3970            ;LENGTH OF 60 DEGREE SEG (USE
EDA9 01 F2              FDB     498             ;SFMAX
EDAB 00 80              FDB     128             ;SFMIN
              *PULSE WIDTHS
EDAD 01 69              FDB     361         ;DOM
EDAF 01 3F              FDB     319
EDB1 01 82              FDB     386         ;DOM
EDB3 01 08              FDB     264
EDB5 01 8B              FDB     395         ;DOM
EDB7 00 C5              FDB     197
              *FREQUENCY TABLE FOR 43 HZ.
              *
EDB9 05       FRQ43     FCB     5           ;NO.-PULSE PERIODS IN 60 DEG SEG
EDBA 03 07              FDB     775             ;LENGTH OF PULSE PERIOD (USEC
EDBC 0F 23              FDB     3875            ;LENGTH OF 60 DEGREE SEG (USEC)
EDBE 01 E5              FDB     485             ;SFMAX
EDC0 00 80              FDB     128             ;SFMIN
              *PULSE WIDTHS
EDC2 01 69              FDB     361         ;DOM
EDC4 01 3F              FDB     319
EDC6 01 82              FDB     386         ;DOM
EDC8 01 08              FDB     264
EDCA 01 8B              FDB     395         ;DOM
EDCC 00 C5              FDB     197
              *FREQUENCY TABLE FOR 44 HZ.
              *
EDCE 05       FRQ44     FCB     5           ;NO.-PULSE PERIODS IN 60 DEG SEG
EDCF 02 F6              FDB     758             ;LENGTH OF PULSE PERIOD (USEC
```

```
SEQ  LOC  OBJ      SOURCE

0881 EDD1 0E CE            FDB    3790           ;LENGTH OF 60 DEGREE SEG (USEC)
0882 EDD3 01 DA            FDB    474            ;SFMAX
0883 EDD5 00 80            FDB    128            ;SFMIN
0884               *PULSE WIDTHS
0885 EDD7 01 69            FDB    361       ;DOM
0886 EDD9 01 3F            FDB    319
0887 EDDB 01 82            FDB    386       ;DOM
0888 EDDD 01 08            FDB    264
0889 EDDF 01 8B            FDB    395       ;DOM
0890 EDE1 00 C5            FDB    197
0891               *FREQUENCY TABLE FOR 45 HZ.
0892               *
0893 EDE3 05       FRQ45   FCB    5         ;NO. -PULSE PERIODS IN 60 DEG SEG
0894 EDE4 02 E5            FDB    741            ;LENGTH OF PULSE PERIOD (USEC)
0895 EDE6 0E 79            FDB    3705           ;LENGTH OF 60 DEGREE SEG (USEC)
0896 EDE8 01 CF            FDB    463            ;SFMAX
0897 EDEA 00 80            FDB    128            ;SFMIN
0898               *PULSE WIDTHS
0899 EDEC 01 69            FDB    361       ;DOM
0900 EDEE 01 3F            FDB    319
0901 EDF0 01 82            FDB    386       ;DOM
0902 EDF2 01 08            FDB    264
0903 EDF4 01 8B            FDB    395       ;DOM
0904 EDF6 00 C5            FDB    197
0905               *FREQUENCY TABLE FOR 46 HZ.
0906               *
0907 EDF8 05       FRQ46   FCB    5         ;NO. -PULSE PERIODS IN 60 DEG SEG
0908 EDF9 02 D5            FDB    725            ;LENGTH OF PULSE PERIOD (USEC)
0909 EDFB 0E 29            FDB    3625           ;LENGTH OF 60 DEGREE SEG (USEC
0910 EDFD 01 C5            FDB    453            ;SFMAX
0911 EDFF 00 80            FDB    128            ;SFMIN
0912               *PULSE WIDTHS
0913 EE01 01 69            FDB    361       ;DOM
0914 EE03 01 3F            FDB    319
0915 EE05 01 82            FDB    386       ;DOM
0916 EE07 01 08            FDB    264
0917 EE09 01 8B            FDB    395       ;DOM
0918 EE0B 00 C5            FDB    197
0919               *FREQUENCY TABLE FOR 47 HZ.
0920               *
0921 EE0D 05       FRQ47   FCB    5         ;NO. -PULSE PERIODS IN 60 DEG SEG
0922 EE0E 02 C5            FDB    709            ;LENGTH OF PULSE PERIOD (USEC)
0923 EE10 0D D9            FDB    3545           ;LENGTH OF 60 DEGREE SEG (USEC
0924 EE12 01 BB            FDB    443            ;SFMAX
0925 EE14 00 80            FDB    128            ;SFMIN
0926               *PULSE WIDTHS
0927 EE16 01 69            FDB    361       ;DOM
0928 EE18 01 3F            FDB    319
0929 EE1A 01 82            FDB    386       ;DOM
0930 EE1C 01 08            FDB    264
0931 EE1E 01 8B            FDB    395       ;DOM
0932 EE20 00 C5            FDB    197
0933               *FREQUENCY TABLE FOR 48 HZ.
0934               *
0935 EE22 05       FRQ48   FCB    5         ;NO. -PULSE PERIODS IN 60 DEG SEG
```

```
SEQ  LOC  OBJ      SOURCE

0936 EE23 02 B6             FDB    694               ;LENGTH OF PULSE PERIOD (USEC
0937 EE25 0D 8E             FDB    3470              ;LENGTH OF 60 DEGREE SEG (USEC
0938 EE27 01 B1             FDB    433               ;SFMAX
0939 EE29 00 80             FDB    128               ;SFMIN
0940                 *PULSE WIDTHS
0941 EE2B 01 69             FDB    361          ;DOM
0942 EE2D 01 3F             FDB    319
0943 EE2F 01 82             FDB    386          ;DOM
0944 EE31 01 08             FDB    264
0945 EE33 01 8B             FDB    395          ;DOM
0946 EE35 00 C5             FDB    197
0947                 *FREQUENCY TABLE FOR 49 HZ.
0948                 *
0949 EE37 05        FRQ49   FCB    5            ;NO.-PULSE PERIODS IN 60 DEG SEG
0950 EE38 02 A8             FDB    680               ;LENGTH OF PULSE PERIOD (USEC)
0951 EE3A 0D 48             FDB    3400              ;LENGTH OF 60 DEGREE SEG (USEC)
0952 EE3C 01 A8             FDB    424               ;SFMAX
0953 EE3E 00 80             FDB    128               ;SFMIN
0954                 *PULSE WIDTHS
0955 EE40 01 69             FDB    361          ;DOM
0956 EE42 01 3F             FDB    319
0957 EE44 01 82             FDB    386          ;DOM
0958 EE46 01 08             FDB    264
0959 EE48 01 8B             FDB    395          ;DOM
0960 EE4A 00 C5             FDB    197
0961                 *FREQUENCY TABLE FOR 50 HZ.
0962                 *
0963 EE4C 05        FRQ50   FCB    5            ;NO.-PULSE PERIODS IN 60 DEG SEG
0964 EE4D 02 9B             FDB    667               ;LENGTH OF PULSE PERIOD (USEC)
0965 EE4F 0D 07             FDB    3335              ;LENGTH OF 60 DEGREE SEG (USE
0966 EE51 01 9F             FDB    415               ;SFMAX
0967 EE53 00 80             FDB    128               ;SFMIN
0968                 *PULSE WIDTHS
0969 EE55 01 69             FDB    361          ;DOM
0970 EE57 01 3F             FDB    319
0971 EE59 01 82             FDB    386          ;DOM
0972 EE5B 01 08             FDB    264
0973 EE5D 01 8B             FDB    395          ;DOM
0974 EE5F 00 C5             FDB    197
0975                 *FREQUENCY TABLE FOR 51 HZ.
0976                 *
0977 EE61 05        FRQ51   FCB    5            ;NO.-PULSE PERIODS IN 60 DEG SEG
0978 EE62 02 8E             FDB    654               ;LENGTH OF PULSE PERIOD (USEC)
0979 EE64 0C C6             FDB    3270              ;LENGTH OF 60 DEGREE SEG (USE
0980 EE66 01 97             FDB    407               ;SFMAX
0981 EE68 00 80             FDB    128               ;SFMIN
0982                 *PULSE WIDTHS
0983 EE6A 01 69             FDB    361          ;DOM
0984 EE6C 01 3F             FDB    319
0985 EE6E 01 82             FDB    386          ;DOM
0986 EE70 01 08             FDB    264
0987 EE72 01 8B             FDB    395          ;DOM
0988 EE74 00 C5             FDB    197
0989                 *FREQUENCY TABLE FOR 52 HZ.
0990                 *
```

```
SEQ  LOC  OBJ       SOURCE

0991 EE76 05        FRQ52   FCB     5           ;NO.-PULSE PERIODS IN 60 DEG SEG
0992 EE77 02 81             FDB     641             ;LENGTH OF PULSE PERIOD (USE
0993 EE79 0C 85             FDB     3205            ;LENGTH OF 60 DEGREE SEG (US
0994 EE7B 01 8F             FDB     399             ;SFMAX
0995 EE7D 00 80             FDB     128             ;SFMIN
0996                *PULSE WIDTHS
0997 EE7F 01 69             FDB     361         ;DOM
0998 EE81 01 3F             FDB     319
0999 EE83 01 82             FDB     386         ;DOM
1000 EE85 01 08             FDB     264
1001 EE87 01 8B             FDB     395         ;DOM
1002 EE89 00 C5             FDB     197
1003                *FREQUENCY TABLE FOR 53 HZ.
1004                *
1005 EE8B 05        FRQ53   FCB     5           ;NO.-PULSE PERIODS IN 60 DEG SEG
1006 EE8C 02 75             FDB     629             ;LENGTH OF PULSE PERIOD (USEC
1007 EE8E 0C 49             FDB     3145            ;LENGTH OF 60 DEGREE SEG (USE
1008 EE90 01 87             FDB     391             ;SFMAX
1009 EE92 00 80             FDB     128             ;SFMIN
1010                *PULSE WIDTHS
1011 EE94 01 69             FDB     361         ;DOM
1012 EE96 01 3F             FDB     319
1013 EE98 01 82             FDB     386         ;DOM
1014 EE9A 01 08             FDB     264
1015 EE9C 01 8B             FDB     395         ;DOM
1016 EE9E 00 C5             FDB     197
1017                *FREQUENCY TABLE FOR 54 HZ.
1018                *
1019 EEA0 05        FRQ54   FCB     5           ;NO.-PULSE PERIODS IN 60 DEG SEG
1020 EEA1 02 69             FDB     617             ;LENGTH OF PULSE PERIOD (USEC
1021 EEA3 0C 0D             FDB     3085            ;LENGTH OF 60 DEGREE SEG (USE
1022 EEA5 01 7F             FDB     383             ;SFMAX
1023 EEA7 00 80             FDB     128             ;SFMIN
1024                *PULSE WIDTHS
1025 EEA9 01 69             FDB     361         ;DOM
1026 EEAB 01 3F             FDB     319
1027 EEAD 01 82             FDB     386         ;DOM
1028 EEAF 01 08             FDB     264
1029 EEB1 01 8B             FDB     395         ;DOM
1030 EEB3 00 C5             FDB     197
1031                *FREQUENCY TABLE FOR 55 HZ.
1032                *
1033 EEB5 05        FRQ55   FCB     5           ;NO.-PULSE PERIODS IN 60 DEG SEG
1034 EEB6 02 5E             FDB     606             ;LENGTH OF PULSE PERIOD (USEC
1035 EEB8 0B D6             FDB     3030            ;LENGTH OF 60 DEGREE SEG (USE
1036 EEBA 01 78             FDB     376             ;SFMAX
1037 EEBC 00 80             FDB     128             ;SFMIN
1038                *PULSE WIDTHS
1039 EEBE 01 69             FDB     361         ;DOM
1040 EEC0 01 3F             FDB     319
1041 EEC2 01 82             FDB     386         ;DOM
1042 EEC4 01 08             FDB     264
1043 EEC6 01 8B             FDB     395         ;DOM
1044 EEC8 00 C5             FDB     197
1045                *FREQUENCY TABLE FOR 56 HZ.
```

```
LOC  OBJ        SOURCE

                *
EECA 03         FRQ56   FCB     3          ;NO.-PULSE PERIODS IN 60 DEG SEG
EECB 03 E0              FDB     992            ;LENGTH OF PULSE PERIOD (USEC)
EECD 0B A0              FDB     2976           ;LENGTH OF 60 DEGREE SEG (USE
EECF 01 79              FDB     377            ;SFMAX
EED1 00 80              FDB     128            ;SFMIN
                *PULSE WIDTHS
EED3 02 69              FDB     617        ;DOM
EED5 01 F7              FDB     503
EED7 02 90              FDB     656        ;DOM
EED9 01 48              FDB     328
                *FREQUENCY TABLE FOR 57 HZ.
                *
EEDB 03         FRQ57   FCB     3          ;NO.-PULSE PERIODS IN 60 DEG SEG
EEDC 03 CF              FDB     975            ;LENGTH OF PULSE PERIOD (USEC)
EEDE 0B 6D              FDB     2925           ;LENGTH OF 60 DEGREE SEG (USEC)
EEE0 01 72              FDB     370            ;SFMAX
EEE2 00 80              FDB     128            ;SFMIN
                *PULSE WIDTHS
EEE4 02 69              FDB     617        ;DOM
EEE6 01 F7              FDB     503
EEE8 02 90              FDB     656        ;DOM
EEEA 01 48              FDB     328
                *FREQUENCY TABLE FOR 58 HZ.
                *
EEEC 03         FRQ58   FCB     3          ;NO.-PULSE PERIODS IN 60 DEG SEG
EEED 03 BE              FDB     958            ;LENGTH OF PULSE PERIOD (USEC
EEEF 0B 3A              FDB     2874           ;LENGTH OF 60 DEGREE SEG (USE
EEF1 01 6C              FDB     364            ;SFMAX
EEF3 00 80              FDB     128            ;SFMIN
                *PULSE WIDTHS
EEF5 02 69              FDB     617        ;DOM
EEF7 01 F7              FDB     503
EEF9 02 90              FDB     656        ;DOM
EEFB 01 48              FDB     328
                *FREQUENCY TABLE FOR 59 HZ.
                *
EEFD 03         FRQ59   FCB     3          ;NO.-PULSE PERIODS IN 60 DEG SEG
EEFE 03 AE              FDB     942            ;LENGTH OF PULSE PERIOD (USEC)
EF00 0B 0A              FDB     2826           ;LENGTH OF 60 DEGREE SEG (USEC)
EF02 01 65              FDB     357            ;SFMAX
EF04 00 80              FDB     128            ;SFMIN
                *PULSE WIDTHS
EF06 02 69              FDB     617        ;DOM
EF08 01 F7              FDB     503
EF0A 02 90              FDB     656        ;DOM
EF0C 01 48              FDB     328
                *FREQUENCY TABLE FOR 60 HZ.
                *
EF0E 03         FRQ60   FCB     3          ;NO.-PULSE PERIODS IN 60 DEG SEG
EF0F 03 9E              FDB     926            ;LENGTH OF PULSE PERIOD (USEC
EF11 0A DA              FDB     2778           ;LENGTH OF 60 DEGREE SEG (USE
EF13 01 5F              FDB     351            ;SFMAX
EF15 00 80              FDB     128            ;SFMIN
                *PULSE WIDTHS
```

```
SEQ  LOC  OBJ     SOURCE

1101 EF17 02 69          FDB    617          ;DOM
1102 EF19 01 F7          FDB    503
1103 EF1B 02 90          FDB    656          ;DOM
1104 EF1D 01 48          FDB    328
1105                     END

1105 LINES ASSEMBLED,  LOC = EF1F,  0000 ERRORS DETECTED.

END ASM V1.1
```

```
SEQ  LOC  OBJ     SOURCE

                  NAM    MCLS9
              *************************************************
              * MCL - MAIN CONTROL LOOP FOR PULSE WIDTH MODULATION *
              *                                                   *
              *************************************************
              *
              * THIS MODULE CONTAINS THE MAIN CONTROL LOOP TO
              * PERFORM THE FOLLOWING FUNCTIONS
              *
              *   . INITIALISATION AT PWR ON
              *   . COMPUTATION OF E*
              *   . OVERLOAD DETECTION OF FEEDBACK
              *   . KEYBOARD MONITOR FOR MANUAL CONTROL
              *   . RAMP UP/ RAMP DOWN IN FREQ.
              *   . LOW VOLTAGE B+ DETECTION
              *
              * THIS MODULE IS EXECUTED ONLY DURING TIME MPU IS
              * ' WAITING ' ON AN INTERRUPT.
              *
              . INTERRUPTS OCCUR EVERY PULSE PERIOD AND VECTOR
              * PROCESSING TO PTIS ( $E000 )
              * THIS SCHEME IS ADOPTED FOR ALL PWM MODIFICATION
              * EVERY PULSE PERIOD BASED ON B+
              *
              * MEMORY LOCATIONS USED BY THIS MODULE
              *
              * (GLOBAL)
         0001 MSTOP   EQU   $0001     ;UNIT STOPPED INDIC. (1=STOPPED)
         0002 MSTART  EQU   $0002     ;START UNIT CMND (1= TRY TO START
         0003 OVRLD   EQU   $0003     ;OVERLOAD FLAG(1= OVERLOAD EXISTS
         0004 INHIB   EQU   $0004     ;RESTART INHIBIT FLAG(1= INHIBIT)
         0044 PIACMD  EQU   $0044     ;PIA CMD (KEEPS FF PR/CLR:SC-PB2)
         0046 FCMD    EQU   $0046     ;COMMANDED OUTPUT FREQUENCY
         004C FOUT    EQU   $004C     ;PRESENT OUTPUT FREQUENCY
         003A VSRAW   EQU   $003A     ;V* INPUT FROM FROM MPU B
         004A BPRAW   EQU   $004A     ;B+ FROM A/D
         0038 FCRAW   EQU   $0038     ;SPEED CMD FROM A/D NOT DETECTED
         0020 DOMID   EQU   $0020     ;DOMINANT'S ID (0=A,1=B,2=C)
         0022 DPF     EQU   $0022     ;DOMINANT'S POLARITY ( 0=+, FF=-)
         0024 J       EQU   $0024     ;FLAG/WAY TO USE COMPL.
         00C0 F60DEG  EQU   $00C0     ;START 60 DEG SEG FLAG(1=START)
              *                    (+=1ST ODD/2ND EVEN,-=1ST EVEN/2ND ODD)
         00C8 FTINCR  EQU   $00C8     ;FREQ TAB POSN.(+4=1ST HALF,-4=2N
         0051 MONFG   EQU   $0051     ;MON CONNECTED FLAG (1=CONN, 0=NO
         00C5 NPPCT   EQU   $00C5     ;NO. OF P.P.'S LEFT IN SEGMENT
         00E6 NORFG   EQU   $00E6     ;NORM OPER. FLAG (0=USE NORM,1=KB
         00E7 VSFLG   EQU   $00E7     ;V* FLAG (0=USE MPB,1=FROM KBD)
         E000 PTIS    EQU   $E000     ;TIMER INTERRUPT SERVICE ROUTINE
         00C1 FTPNW   EQU   $00C1     ;NEW FREQUENCY TABLE POINTER
         0006 OVLD1   EQU   $0006     ;OVRLD COND FLAG(1=OVLD,10%<OL<1
         00E8 INDEX   EQU   $E8       ;FREQ TABLE INDEX MS OF ADDRESS
         00C3 FTPTR   EQU   $00C3     ;FREQ TAB POINTER-PRESENT POSITI
         6000 ACIAC   EQU   $6000     ;ACIA CONTROL REGISTER
```

```
SEQ   LOC   OBJ     SOURCE

0055        0009 DIRCMD  EQU    $0009        ;CCW DIR CMD FLG (1=CCW)
0056        0010 DIROUT  EQU    $0010        ;CCW DIR OUT FLG (1=CCW)
0057             *
0058             * (LOCALS)
0059             *
0060        0050 BPFLG   EQU    $0050        ;B+ FLAG (0=B+<80, 1=B+>80)
0061        0048 VSTAR   EQU    $0048        ;VSTAR CONTROLLED BY KEYBOARD
0062        00EA ESOC    EQU    $00EA        ;ADDRESS OF E* OVERLOAD CALC VALUE
0063        0005 K       EQU    $05          ;K=32; FILTER CONSTANT, FC=XXX HZ
0064        00F0 AVE     EQU    $00F0        ;ADDRESS LOC FOR FILTER YN
0065        0001 ESMM    EQU    $01          ;E* MAX MS VALUE (195)
0066        0000 ESML    EQU    $00          ;E* MAX LS
0067        0000 ES1MM   EQU    $00          ;E* LOWER MAX MS (65)
0068        0080 ES1ML   EQU    $80          ;E*    "    "  LS
0069        003C ESTAR   EQU    $003C        ;E* VALUE - NOW
0070        00F0 YN      EQU    AVE          ;E* AVE PREVIOUS CALCULATION
0071        00CA FTPNT   EQU    $00CA        ;TEMP LOC FOR NEW POINTER ADDR
0072        0040 ESRT    EQU    $0040        ;E* TEMP LOC FOR SF (1+V*/8)
0073        00F0 OVCMP   EQU    $00F0        ;OVERLOAD COMPARISON VALUE
0074        00D8 VPADDR  EQU    $00D8        ;V* PEAK TABLE POINTER ADDR
0075        00DB VSREF   EQU    $00DB        ;V* MAX FOR "F"
0076        00EF OUTCTL  EQU    $EF          ;V* OUT OF CONTROL - LS
0077        000D OUTCTM  EQU    $0D          ;V* OUT OF CONTROL - MS
0078        005A SHTDN   EQU    $005A        ;ADDR SHUT DOWN FLAG
0079             *
0080             *
0081             * (I/O ADDRESSES)
0082             *
0083        5000 PIA     EQU    $5000        ;PIA ADDRESS
0084        4000 PTM1    EQU    $4000        ;PTM1 ADDRESS
0085        3000 PTM2    EQU    $3000        ;PTM2 ADDRESS
0086        2007 WLATCH  EQU    $2007        ;' W ' LATCH FOR MPB
0087        6000 ACIAS   EQU    $6000        ;ACIA STATUS REG
0088        6001 ACIAD   EQU    $6001        ;ACIA DATA REG
0089             *
0090             *
0091             *
0092        0001         LSYMB  1
```

```
SEQ  LOC  OBJ      SOURCE

                   *
                   *********************************
                   * PROGRAM INITIALIZATION ROUTINE *
                   * *******************************
                   *
          D800            ORG    $D800
                   *
                   * ZERO RAM -- LOCS 0 THRU $03FF
                   * PRESERVE MONITOR CONNECTED FLAG  $0051
                   *          NORMAL OP FLAG          $00E6
                   *          V* FLAG                 $00E7
                   *
     D800 0F       INIT   SEI                  ;DISABLE INTERRUPTS
     D801 96 51           LDA A  MONFG           ;LOAD MONITOR CONN FLAG
     D803 DE E6           LDX    NORFG             ;LOAD NORM OP & V* FLAGS
     D805 B7 FFED         STA A  $FFED           ;TEMP STORE FLAG
     D808 FF FFEE         STX    $FFEE           ;TEMP STORE FLAGS
     D80B 4F              CLR A                ;SET CLR VALUE
     D80C CE 0000         LDX    #0                ;LOAD BOTTOM ADDRESS
     D80F A7 00    INIT1  STA A  00,X            ;CLEAR RAM LOC
     D811 08              INX                  ;CHANGE ADDRESS
     D812 8C 03FF         CPX    #$03FF            ;TEST FOR LAST ADDRESS DONE
     D815 26 F8           BNE    INIT1             ;SKIP IF MORE ADDRESSES
     D817 B6 FFED         LDA A  $FFED           ;BRING FLAG BACK
     D81A CE FFEE         LDX    #$FFEE            ;BRING FLAGS BACK
     D81D 97 51           STA A  MONFG           ;RESTORE MON CONN FLAG
     D81F DF E6           STX    NORFG             ;RESTORE NORM OP & V* FLAGS
                   *
                   * SET THE STACK POINTER
                   *
     D821 8E FF00         LDS    #$FF00
                   *
                   * SET IRQ POINTER TO THE START OF THE PULSE TIMER
                   * MODULE INTERRUPT SERVICE ROUTINE
                   *
     D824 CE E000         LDX    #PTIS             ;GET VALUE
     D827 FF FFF8         STX    $FFF8             ;SET IRQ VECTOR
                   *
                   * INITIALIZE THE TIMERS AND SET THE UNIT STOPPED FLAG
                   *
     D82A BD DAB0         JSR    TRESET        ;TIMER RESET ROUTINE
     D82D 86 01           LDA A  #1
     D82F 97 01           STA A  MSTOP         ;SET UNIT STOPPED FLAG
                   *
                   * SET OVERLOAD INDICATOR OFF(VIA PIA) BY "PMON" ROUTINE
                   *   PIA HAS BEEN SET UP PA = ALL INPUTS
                   *                       PB = 6 OUTPUTS + 1 INPUT
                   *
     D831 86 34           LDA A  #$34          ;SET OC WORD FORMAT
     D833 B7 5003         STA A  PIA+3         ;TURN OL OFF THRU CRB OF PIA
                   * ENABLE INTERRUPTS AND ENTER THE MAIN CONTROL LOOP
                   *
     D836 0E              CLI
     D837 20 47           BRA    MCL
```

```
SEQ  LOC  OBJ       SOURCE

                    * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
                    * MCL--MAIN CONTROL LOOP FOR THE PWM UNIT CONTROLLER *
                    * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
                    *
          D880              ORG    $D880
                    *
                    *
                    * FIRST TEST IF THE UNIT IS STOPPED
                    *
     D880 96 01     MCL     LDA A  MSTOP        ;GET UNIT STOPPED INDIC
     D882 27 17             BEQ    MCL02        ;SKIP IF RUNNING
                    *
                    *
                    *    UNIT START-UP CONTROL
                    *    ----------------------
                    *
                    * THE UNIT IS STOPPED.  CHECK THE FREQUENCY COMMAND
                    * TO SEE IF THE OPERATOR WANTS IT STARTED.
                    *   INPUT A COMMAND FROM KEYBOARD OR READ AND CONVERT
                    *   SPDCMD FROM A/D
     D884 BD DCA0           JSR    ADIN         ;INPUT FCMD FROM A/D
     D887 BD DCE3           JSR    CSPDC        ;INPUT STORE FCMD
     D88A 96 46             LDA A  FCMD         ;TEST FOR NON-ZERO
     D88C 26 05             BNE    MCL01        ;IF NON-ZERO,TRY TO START
                    *
                    *
                    *   FREQUENCY COMMAND IS EQUAL TO 0.  CLEAR THE RESTART
                    *   INHIBIT FLAG TO ALLOW UNIT STARTUP.  THE RESTART
                    *   INHIBIT FLAG IS SET BY THE OVERLOAD PROTECTION
                    *   ROUTINE IF AN OVERLOAD CONDITION HAS FORCED
                    *   UNIT SHUTDOWN.  THE OPERATOR MUST COMMAND "OFF"(FCMD=0)
                    *   TO CLEAR THE FLAG AND THEN COMMAND THE UNIT TO START
                    *
     D88E 7F 0004           CLR    INHIB        ;CLEAR THE INHIBIT FLAG
     D891 20 ED             BRA    MCL          ;SKIP BACK TO TRY AGAIN
                    *
                    *
                    *   OPERATOR HAS COMMANDED UNIT STARTUP.  TEST THE RESTART
                    *   INHIBIT FLAG TO SEE IF WE WILL LET HIM.
                    *   IF STARTUP IS O.K.  SET THE START UNIT FLAG
                    *
     D893 96 04     MCL01   LDA A  INHIB        ;UNIT RESTART INHIBIT FLAG
     D895 26 E9             BNE    MCL          ;BRANCH BACK IF SET
     D897 86 01             LDA A  #1           ;ELSE, SET START UNIT FLAG
     D899 97 02             STA A  MSTART       ;SET UNIT START FLAG
                    *
                    *  UNIT IS RUNNING
                    *
     D89B BD DCE3   MCL02   JSR    CSPDC        ;IN/CONVERT FCMD,B+,V*
                    *
```

```
SEQ  LOC  OBJ      SOURCE

0200                *
0201                *
0202                *
0203                * CALL THE OVERLOAD PROTECTION ROUTINE TO PROTECT
0204                * THE UNIT IN CASE OF OVERLOAD
0205                *   TESTS V* TO DETERMINE IF OVERLOAD EXISTS
0206                *   TESTS B+ TO DETERMINE IF TOO LOW (B+ <80 VOLTS)
0207
0208                *
0209 D89E BD DC00          JSR    PROTCT                ;OPERATION PROTECT
0210                *
0211                * TEST PROTECT FLAGS TO DETERMINE IF FREQUENCY
0212                *  SHOULD CHANGE
0213                *
0214                *  IF MAX OVERLOAD EXISTS ------ RAMPDOWN
0215                *  IF B+ < 80 V OR B+ > 140 V -- RAMPDOWN
0216                *  IF OVERLOAD CONDITION 1  ---- DON'T CHANGE FREQUENCY
0217                *
0218                *  IF NO PROTECT FLAGS TEST FREQUENCY
0219                *  FOUT < FCMD    -------------- RAMPUP
0220                *  FOUT > FCMD    -------------- RAMPDOWN
0221                *  FOUT = FCMD    -------------- DON'T CHANGE FREQUENCY
0222                *
0223                *  IF FCMD = 0 THEN RAMP DOWN AT THE NORMAL RATE
0224                *
0225 D8A1 96 46            LDA A  FCMD         ;LD FREQ COMMAND
0226 D8A3 27 12            BEQ    MCL03        ;SKIP IF ZERO
0227                *
0228                * MAX OVERLOAD TEST
0229                *
0230 D8A5 96 03            LDA A  OVRLD        ;OVERLOAD INDICATOR
0231 D8A7 26 56            BNE    OLRMPD       ;SKIP FOR OVERLOAD RAMPDOWN
0232                *
0233                *
0234                *TEST TO SEE IF B+ INPUT IS LOW (<80 V) OR HIGH (>140 V)
0235                * IF IT IS,WE MUST TURN OFF BEFORE THE POWER SUPPLY
0236                * CRASHES.
0237                *
0238 D8A9 96 50            LDA A  BPFLG        ;LD B+ FLAG
0239                * IF SET, RAMPDOWN
0240 D8AB 26 0A            BNE    MCL03        ;BRANCH IF B+<80 OR >140V
0241                *
0242                * DETERMINE IF WE'RE AT COMMANDED OUTPUT FREQUENCY
0243                * COMPARE THE COMMANDED UNIT SPEED WITH THE ACTUAL
0244                * OUTPUT FREQUENCY  - IF LOW  RAMPUP
0245                *                     IF HIGH RAMPDOWN
0246                *                     IF THERE,DO NOTHING
0247                *
0248 D8AD 96 46            LDA A  FCMD         ;COMMANDED FREQUENCY
0249 D8AF 90 4C            SUB A  FOUT         ;SUBTRACT OUTPUT FREQUENCY
0250 D8B1 27 1C            BEQ    TFOUT        ;SKIP IF THERE(NO F CHANGE)
0251 D8B3 2B 02            BMI    MCL03        ;SKIP IF HIGH (GO DOWN IN F)
0252 D8B5 20 11            BRA    MCL04        ;SKIP TO RAMP UP
```

```
LOC  OBJ       SOURCE

                   *
                   *DECISION MADE TO GO DOWN IN FREQUENCY
                   *  WE GOT HERE BECAUSE  1. FCMD < FOUT
                   *                    OR  2. OVERLOAD EXISTS
                   *                    OR  3. B+ < 80 VOLTS
D8B7 BD DB80 MCL03   JSR    RAMPDN        ;RAMP DOWN FOUT
D8BA 96 4C   MCL3A   LDA A  FOUT          ;LD OUTPUT FREQ
D8BC 26 11           BNE    TFOUT         ;SKIP IF NOT "0"
                   *IF FOUT =0 FORCED BY OVERLOAD, SET RESTART INHIBIT FLAG
                   *
D8BE 96 03           LDA A  OVRLD         ;GET FLAG
D8C0 27 0D           BEQ    TFOUT         ;SKIP IF OVRLD DIDN'T CAUSE
D8C2 86 01           LDA A  #1            ;SET FLAG VALUE
D8C4 97 04           STA A  INHIB         ;SET FLAG
D8C6 20 07           BRA    TFOUT         ;SEE IF FOUT=0
                   *
                   *DECISION MADE TO GO UP IN FREQUENCY
                   * WE GOT HERE BECAUSE    1. FCMD > FOUT
                   *                   AND  2. OVERLOAD DOESN'T EXIST
                   *                   AND  3. B+ > 80 VOLTS
                   *  MUST VERIFY PARTIAL OVERLOAD DOESN'T EXIST
                   *
D8C8 96 06   MCL04   LDA A  OVLD1         ;LD FLAG
D8CA 26 03           BNE    TFOUT         ;SKIP IF PARTIAL OVERLOAD
D8CC BD DB00         JSR    RAMPUP        ;ELSE, RAMP UP OUTPUT FREQ
                   *
                   *
                   * NEXT STEP AFTER CHANGING FOUT IS TO ADJUST THE FREQUENCY
                   * TABLE POINTER IF WE'RE NOT AT "0" FCMD
                   *
                   * TEST THE OUTPUT FREQUENCY.
                   *
                   *
D8CF 96 4C   TFOUT   LDA A  FOUT          ;OUTPUT FREQUENCY
D8D1 27 23           BEQ    STUNT         ;IF=0, STOP THE UNIT
                   *
                   * THIS SECTION FINDS & STORES THE ADDRESS OF THE FREQUENCY
                   *    TABLE INDEX CORRESPONDING TO THE OUTPUT FREQUENCY
                   *    HENCE POINTS TO LOCATION FOR PW PARAMETERS (PTIS USES)
                   *
D8D3 D6 4C   CHPTR   LDA B  FOUT          ;OUTPUT FREQ
D8D5 C0 06           SUB B  #6            ;GET RID OF OFFSET
D8D7 2A 02           BPL    AINDX         ;SKIP IF > 6 HZ
D8D9 C6 00           LDA B  #00           ;SET TO MIN F=6
D8DB 58      AINDX   ASL B                ;ACCOUNT FOR 2 BYTES/F
D8DC 86 E8           LDA A  #INDEX        ;LD MS OF INDEX
D8DE D7 CB           STA B  FTPNT+1       ;STORE LS OF ADDR
D8E0 97 CA           STA A  FTPNT         ;STORE MS OF ADDR
                   * CHANGE ADDRESS POINTER NOW THAT ITS' GENERATED
D8E2 DE CA           LDX    FTPNT         ;LD NEW POINTER
D8E4 EE 00           LDX    0,X           ;GET TABLE ADDRESS
D8E6 DF C1           STX    FTPNW         ;STORE NEW FT POINTER
                   *
                   * IF THE START UNIT FLAG IS SET, START THE UNIT FLAG
```

```
SEQ  LOC  OBJ      SOURCE

0308               *
0309 D8E8 96 02            LDA A  MSTART      ;START UNIT COMMAND FLAG
0310 D8EA 27 94            BEQ    MCL         ;IF ZERO, DON'T START
0311               *
0312               *    START UNIT LOGIC
0313               *    ----------------
0314               *
0315 D8EC BD DA20          JSR    TSTART      ;TIMER STARTUP ROUTINE
0316 D8EF 4F               CLR A
0317 D8F0 97 02            STA A  MSTART      ;CLR START UNIT FLAG
0318 D8F2 97 01            STA A  MSTOP       ;CLR UNIT STOPPED INDIC
0319 D8F4 20 8A    MCLCN   BRA    MCL
0320               *
0321               *
0322               *    STOP UNIT LOGIC
0323               *    ---------------
0324               *
0325 D8F6 BD DAB0  STUNT   JSR    TRESET      ;TIMER RESET ROUTINE
0326 D8F9 7E DDA6          JMP    STROUT      ;STOP ROUTINE
0327 D8FC 01               NOP
0328 D8FD 20 F5            BRA    MCLCN       ;BRA TO MCL CONNECT
0329               *
0330               * MAX OVERLOAD - RAMPDOWN SLOWLY
0331               *
0332               * IF SHUT DOWN FLAG SET - FORCE FOUT = 0
0333               *
0334               *
0335 D8FF 86 00    OLRMPD  LDA A  #$00
0336 D901 27 05            BEQ    OLRMP1      ;SKIP IF IN CONTRL
0337 D903 7F 004C          CLR    FOUT        ;SET FOUT=0
0338 D906 20 B2            BRA    MCL3A       ;SKIP TO CONT SHUT DOWN
0339 D908 BD DB91  OLRMP1  JSR    SLWRD       ;CALL RAMP DOWN SLOWLY
0340 D90B 20 AD            BRA    MCL3A       ;SKIP TO TEST FREQ
```

```
SEQ  LOC  OBJ       SOURCE

0342                * * * * * * * * * * * * * * * * *
0343                * TSTART--TIMER STARTUP ROUTINE *
0344                * * * * * * * * * * * * * * * * *
0345                *
0346                * THIS ROUTINE PERFORMS THE LOGIC NECESSARY TO INITIALIZE
0347                * THE TWO PULSE TIMER MODULES (PTM) AND BEGIN THE PWM OUTPUT.
0348                *
0349      DA20              ORG    $DA20
0350                *
0351 DA20 0F        TSTART  SEI                  ;DISABLE INTERRUPTS
0352                *
0353                * SET ALL THREE PTM #1 TIMERS FOR SINGLE SHOT MODE
0354                * AND APPLY INTERNAL RESET (IR)
0355                *
0356                * CONTROL WORD: 1011001X=SINGLE-SHOT MODE, USE ENABLE CLOCK
0357                *           (=B2(HEX))16 BIT MODE,RESET ON GATE .OR. RESET
0358                *                     INTERRUPT DISABLED/OUTPUT ENABLED
0359                *
0360 DA21 86 B2             LDA A  #$B2          ;SET THE CONTROL WORD
0361 DA23 B7 4001           STA A  PTM1+1        ;INIT TIMER #2
0362 DA26 B7 4000           STA A  PTM1          ;INIT TIMER #3
0363 DA29 8A 01             ORA A  #1            ;SET CTRL WD TO ENB #1
0364 DA2B B7 4001           STA A  PTM1+1        ;ENABLE WRITE TO CR 1
0365 DA2E B7 4000           STA A  PTM1          ;INIT TIMER #1 AND APPLY IR
0366                *
0367                * LOAD PTM#1 LATCHES WITH MAX VALUE
0368                *
0369 DA31 CE FFFF           LDX    #$FFFF        ;MAX COUNT
0370 DA34 FF 4002           STX    PTM1+2        ;TO TIMER#1 LATCHES
0371 DA37 FF 4004           STX    PTM1+4        ;TO TIMER#2 LATCHES
0372 DA3A FF 4006           STX    PTM1+6        ;TO TIMER#3 LATCHES
```

```
SEQ  LOC  OBJ       SOURCE

0374                *
0375                * PULSE THE INTERNAL RESET BIT TO LOAD THE COUNTERS
0376                * FROM THE LATCHES
0377                *
0378 DA3D 84 FE             AND A  #$FE         ;REMOVE IR
0379 DA3F B7 4000           STA A  PTM1         ;WRITE IT TO PTM#1
0380 DA42 8A 01             ORA A  #1           ;RE-APPLY IR
0381 DA44 B7 4000           STA A  PTM1         ;WRITE IT TO PTM#1
0382                *
0383                *
0384                *
0385                * INITIALIZE PTM#2 AND APPLY INTERNAL RESET TO PTM#2
0386                *   TIMER #1:  CONTINUOUS MODE, USE ENABLE CLOCK,16 BIT MODE
0387                *              RESET ON GATE .OR. RESET
0388                *              INTERRUPT AND OUTPUT ENABLED
0389                *              (USED AS INTERVAL TIMER FOR PWM)
0390                *
0391                *   TIMER #2:  NOT USED
0392                *
0393                *   TIMER #3   CONTINUOUS MODE, USE ENABLE CLOCK, 16 BIT MODE
0394                *              RESET ON GATE .OR. RESET
0395                *              INTERRUPT DISABLED, OUTPUT ENABLED
0396                *              (USED AS BAUD RATE GENERATOR FOR ACIA)
0397                *
0398                * CONTROL WORD : 1101001X FOR ABOVE CONDITIONS
0399                *                     (=D2(HEX))
0400                *   TIMER #1 - USED AS INTERVAL (PULSE PERIOD LENGTH) FOR PWM
0401                *
0402 DA47 5F                CLR B
0403 DA48 F7 3001           STA B  PTM2+1        ;WRITE CR2 TO SELECT CR3
0404 DA4B C6 92             LDA B  #$92          ;TIMER #3 CONTROL WORD
0405 DA4D F7 3000           STA B  PTM2          ;INIT TIMER #3
0406 DA50 C6 01             LDA B  #1
0407 DA52 F7 3001           STA B  PTM2+1        ;SELECT CR1
0408 DA55 C6 D3             LDA B  #$D3          ;TIMER #1 CONTROL WORD
0409 DA57 F7 3000           STA B  PTM2          ;INIT TIMER#1 AND APPLY IR
0410                *
0411                * LOAD PTM #2 TIMER #1 LATCHES WITH 794 MICROSECOND COUNT
0412                *    COUNT FOR TPP WITH F=6 HZ
0413                *
0414 DA5A CE 031A           LDX    #794
0415 DA5D FF 3002           STX    PTM2+2        ;COUNT TO TIMER#1 LATCHES
0416                *
0417                * PULSE THE INTERNAL RESET (IR) BIT TO LOAD PTM#2 COUNTERS
0418                *
0419                *
0420 DA60 C4 FE             AND B  #$FE          ;REMOVE INTERNAL RESET
0421 DA62 F7 3000           STA B  PTM2          ;WRITE IT TO CR 1
0422 DA65 CA 01             ORA B  #1            ;RE-APPLY IR
0423 DA67 F7 3000           STA B  PTM2          ;WRITE IT TO CR1
```

```
SEQ  LOC  OBJ       SOURCE

0425                *
0426                * SET UP RAM LOCATIONS USED BY PULSE TIMER INTERRUPT SERVICE
0427                * ROUTINE (PTIS) TO START AT BEGINNING OF 60 DEGREE SEGMENT.
0428                *
0429 DA6A 86 01            LDA A   #+1          ;SET FLAG VALUE
0430 DA6C 97 C0            STA A   F60DEG       ;SET FLAG-60 DEG BOUND
0431 DA6E 97 20            STA A   DOMID        ;SET DOM ID =1=PHASE B
0432 DA70 97 22            STA A   DPF          ;SET LAST POLARITY= +
0433 DA72 97 24            STA A   J            ;SET INDEX TO 1ST HALF
0434 DA74 CE 0004          LDX     #+4          ;SET INC FOR FREQ TAB(1ST HALF)
0435 DA77 DF C8            STX     FTINCR       ;SET IN MEM LOC
0436 DA79 86 A6            LDA A   #$A6         ;SET PR/RST FF CMD
0437 DA7B B7 5002          STA A   PIA+2        ;ENB FF'S TO PRESET
0438 DA7E 97 44            STA A   PIACMD       ;SAVE CMD
0439                * SECTION SETS UP FREQUENCY TABLE POINTERS
0440 DA80 CE E880          LDX     #$E880       ;TOP OF TABLE(6 HZ)
0441 DA83 DF C1            STX     FTPNW        ;SET NEW POINTER ADDR
0442 DA85 DF C3            STX     FTPTR        ;SET EXISTING POINTER
0443 DA87 CE E800          LDX     #$E800       ;TOP OF INDEX
0444 DA8A DF CA            STX     FTPNT        ;SET TEMP ADDR
0445                *
0446                *
0447                * REMOVE THE INTERNAL RESETS TO ALLOW BOTH TIMERS TO OPERATE
0448                * RESTORE THE INTERRUPT FLAG AND RETURN
0449                *
0450 DA8C 86 B2            LDA A   #$B2           ;CTRL WD PTM#1 T1
0451 DA8E C6 D2            LDA B   #$D2           ;CTRL WD PTM#2 T1
0452 DA90 B7 4000          STA A   PTM1           ;REMOVE IR FROM PTM#1
0453 DA93 F7 3000          STA B   PTM2           ;REMOVE IR FROM PTM#2
0454                *
0455 DA96 0E               CLI                  ;ENABLE INTERRUPTS
0456 DA97 39               RTS                  ;AND RETURN
```

```
SEQ  LOC  OBJ      SOURCE

                   * * * * * * * * * * * * * * * *
                   * TRESET--TIMER RESET ROUTINE *
                   * * * * * * * * * * * * * * * *
                   *
          DAB0           ORG    $DAB0
                   *
                   * THIS ROUTINE CONTAINS THE LOGIC TO RESET THE TIMERS AND
                   * STOP THE PWM OUTPUT
                   *
     DAB0 0F       TRESET  SEI                  ;DISABLE INTERRUPTS
                   *
                   *
                   * DISABLE OUTPUT OF PTM#2 TIMER #1
                   * ALLOW TIMER #3 TO OPERATE
                   *
     DAB1 86 01          LDA A  #1
     DAB3 B7 3001        STA A  PTM2+1          ;SELECT CR 1
     DAB6 86 12          LDA A  #$12            ;CODE TO DISABLE OUTPUT&IRQ
     DAB8 B7 3000        STA A  PTM2            ;STOP TIMER #1
                   *
                   *
                   * APPLY INTERNAL RESET TO CLEAR PTM#1
                   *
     DABB 86 B3          LDA A  #$B3
     DABD B7 4001        STA A  PTM1+1           ;SELECT CR1
     DAC0 B7 4000        STA A  PTM1             ;APPLY INTERNAL RESET
                   *
                   *
                   * OUTPUT FOUT=0 TO 'W' LATCH
                   *
     DAC3 4F             CLR A
     DAC4 B7 2007        STA A  WLATCH            ;OUTPUT TO W LATCH
                   *
                   * RESTORE INTERRUPT MASK FLAG AND RETURN
                   *
     DAC7 0E             CLI                    ;ENABLE INTERRUPTS
     DAC8 39             RTS
```

```
LOC  OBJ      SOURCE

                    *
                    * RAMP UP ROUTINE
                    *
          DB00            ORG     $DB00
                    *
DB00 D6 4D          RAMPUP  LDA B   FOUT+1      ;LD FOUT LS BYTE
DB02 96 4C                  LDA A   FOUT        ;LD MS OF FOUT
DB04 CB 01                  ADD B   #01         ;INCR FOUT BY +1(WAS 2)
DB06 89 00                  ADC A   #00         ;INCR MS
DB08 81 3C                  CMP A   #60         ;TEST FOR FOUT>60
DB0A 2A 05                  BPL     SMOUT       ;BRANCH IF FOUT>60
DB0C D7 4D          SFOUT   STA B   FOUT+1      ;STORE NEW LS OF FOUT
DB0E 97 4C                  STA A   FOUT        ;STORE FOUT MS BYTE
DB10 39             ERU     RTS                 ;RETURN FROM RAMPUP
DB11 5F             SMOUT   CLR B               ;CLEAR FOUT LS BYTE
DB12 86 3C                  LDA A   #60         ;LD FOUT MS=60
DB14 20 F6                  BRA     SFOUT       ;BRANCH TO STORE FOUT
                    *
                    *
                    * RAMP DOWN ROUTINE
                    *
          DB80            ORG     $DB80
                    *
DB80 D6 4D          RAMPDN  LDA B   FOUT+1      ;LD FOUT LS BYTE
DB82 96 4C                  LDA A   FOUT        ;LD FOUT MS
DB84 C0 01                  SUB B   #01         ;DECR FOUT LS BY -1(WAS 2)
DB86 82 00                  SBC A   #00         ;DECR FOUT MS
DB88 2A 02                  BPL     ERD         ;BRANCH IF FOUT IS +
DB8A 4F                     CLR A               ;LD 0 FOR FOUT
DB8B 5F                     CLR B               ;LD LS OF FOUT
DB8C D7 4D          ERD     STA B   FOUT+1      ;STORE FOUT LS
DB8E 97 4C                  STA A   FOUT        ;STORE 0 FOR FOUT MS
DB90 39                     RTS                 ;RETURN FROM RAMPDN
                    *
                    * SLOW RAMP DOWN ROUTINE - USED ONLY IN OVERLOAD
                    *
DB91 D6 4E          SLWRD   LDA B   FOUT+2      ;LD LS OF FOUT
DB93 96 4D                  LDA A   FOUT+1      ;LD MID BYTE OF FOUT
DB95 C0 80                  SUB B   #$80        ;DECR FOUT LS
DB97 82 00                  SBC A   #$00        ;DECR MID BYTE
DB99 D7 4E                  STA B   FOUT+2      ;SAVE LS FOUT
DB9B 97 4D                  STA A   FOUT+1      ;SAVE MID
DB9D D6 4C                  LDA B   FOUT        ;LD MS OF FOUT
DB9F C2 00                  SBC B   #$00        ;DECR MS OF FOUT
DBA1 2A 07                  BPL     ESRD        ;SKIP TO END SLOW RAMP
DBA3 5F                     CLR B               ;CLR FOUT VALUE
DBA4 7F 004D                CLR     FOUT+1      ;CLR MID
DBA7 7F 004E                CLR     FOUT+2      ;CLR LS
DBAA D7 4C          ESRD    STA B   FOUT        ;SAVE MS OF FOUT
DBAC 39                     RTS                 ;RETURN TO MAIN STREAM
```

```
SEQ  LOC  OBJ       SOURCE

                    *
                    *
                    * * * * * * * * * * * * * *
                    * PROTECTION ROUTINE
                    * * * * * * * * * * * * * *
                    *
          DC00              ORG     $DC00
                    *
                    * THIS SUBROUTINE CONTROLS FLAGS USED BY THE MAIN
                    * PROGRAM CONTROL LOOP TO DETERMINE IF CONTROLLER
                    * IS OPERATING WITHIN DESIRED CONSTRAINTS.
                    *
                    * OPERATIONAL CONSTRAINTS EXAMINED BY THIS SUBROUTINE :
                    *   B+ -- INPUT VOLTAGE
                    *      NORMAL OPERATION 80V < B+ < 140V
                    *      LOW VOLTAGE            B+ < 80V SET FLAG-RAMPDOWN
                    *      HIGH VOLTAGE           B+ > 140V SET FLAG-RAMPDOWN
                    *
                    *   V* -- FEEDBACK ERROR          FLAG   /OL LAMP/FOUT CTRL
                    *    V* > ESMM                     SET      ON     RAMPDOWN
                    *    VSREF< V* < ESMM              RESET    ON     FOUT=FOUT
                    *    V* < VSREF                    RESET   OFF     FOUT=FCMD
                    *
                    * V* IS ERROR READ IN DIGITAL INTERFACE
                    *
     DC00 D6 4A     PROTCT  LDA B   BPRAW       ;B+ RAW DATA
     DC02 C1 50             CMP B   #80         ;B+ - 80 V
     DC04 23 09             BLS     PVOLT       ;SKIP IF TOO LOW
     DC06 C1 B4             CMP B   #$B4        ;B+ - 140 V
     DC08 22 05             BHI     PVOLT       ;SKIP IF TOO HIGH
     DC0A 7F 0050           CLR     BPFLG       ;CLR B+ PROBLEM FLAG
     DC0D 20 04             BRA     ESPRT       ;SKIP FOR E* FILTER
                    *
                    * B+ PROBLEM DETECTED
                    *    B+ IS EITHER LOW OR HIGH.  IN EITHER CASE
                    *    WANT TO CAUSE RAMPDOWN BEFORE POWER SUPPLY
                    *    QUITS.
     DC0F 86 01     PVOLT   LDA A   #1          ;SET FLAG VALUE
     DC11 97 50             STA A   BPFLG       ;SET FLAG
```

```
SEQ  LOC  OBJ      SOURCE

0587               *
0588               *NEXT STEP IS V* (OVERLOAD) PROTECTION
0589               *
0590               * TEST 1ST TO SEE IF WE'RE OUT OF CONTROL
0591 DC13 96 48    ESPRT   LDA A  VSTAR        ;LD V* MS
0592 DC15 D6 49            LDA B  VSTAR+1      ;LD V* LS
0593 DC17 C0 EF            SUB B  #OUTCTL      ;SUBT LS LIMIT
0594 DC19 82 0D            SBC A  #OUTCTM      ;SUBT MS LIMIT
0595 DC1B 2A 51            BPL    OTCT         ;SKIP IF > LIMIT
0596               *DO COMPARISON OF V* TO TABLE VALUE FOR VSREF
0597 DC1D 96 4C            LDA A  FOUT         ;LD OUTPUT FREQUENCY
0598 DC1F 48               ASL A               ;ACCOUNT FOR 2BYTES/F
0599 DC20 C6 E4            LDA B  #$E4         ;LD V* PEAK MS ADDR
0600 DC22 D7 DB            STA B  VSREF        ;STORE ADDR MS
0601 DC24 97 DC            STA A  VSREF+1      ;STORE ADDR LS
0602 DC26 DE DB            LDX    VSREF        ;LD V* REF ADDR
0603               * NOW DO COMPARISON
0604 DC28 86 0C            LDA A  #$0C         ;LD MAX CURRENT
0605 DC2A C6 00            LDA B  #$00
0606 DC2C D0 49            SUB B  VSTAR+1      ;VSREF - V* LS
0607 DC2E 92 48            SBC A  VSTAR        ;VSREF - V* MS
0608 DC30 2A 1F            BPL    NOLC         ;SKIP IF NO OVR LD
```

```
SEQ  LOC  OBJ       SOURCE

0610                *
0611                * DO PROTECT USING COMPARISON VALUE
0612                *
0613 DC32 D7 F1     PRTCT1  STA B  OVCMP+1     ;TEMP STORE LS -COMPARISON
0614 DC34 97 F0             STA A  OVCMP       ;TEMP STORE MS -COMPARISON
0615                * TESTS IDENTIFY RANGE OF V* AVERAGE ERROR
0616                *     1ST TEST - IS IT > ES1MM ?
0617                *
0618 DC36 CB 80             ADD B  #ES1ML      ;CMP - LS LOWER REF
0619 DC38 89 00             ADC A  #ES1MM      ;CMP - MS LOWER REF
0620 DC3A 2A 0A             BPL    OLC1        ;SKIP IF COND 1
0621                * WE HAVE OVERLOAD - BUT HOW BAD
0622                *     2ND TEST - IS IT > ESMM ?
0623                *
0624                *  IF FCMD=0 WITH OVERLOAD (COND 1 OR 2) - SHUTDOWN !
0625                *
0626 DC3C D6 F1             LDA B  OVCMP+1     ;RELOAD LS
0627 DC3E 96 F0             LDA A  OVCMP       ;RELOAD MS
0628 DC40 CB 00             ADD B  #ESML       ;CMP - E* MAX LS
0629 DC42 89 01             ADC A  #ESMM       ;CMP - E* MAX MS
0630 DC44 2B 1E             BMI    OLCM        ;SKIP IF MAX OVERLOAD
0631                * OVERLOAD IS VSREF <  V* < ESMM
0632                *  TURN ON OVERLOAD LAMP,SET OVERLOAD CONDITION 1 FLAG
0633                *    DON'T ALLOW FREQUENCY TO CHANGE (UP OR DOWN)
0634 DC46 86 3C     OLC1    LDA A  #$3C        ;LD OC LAMP ON CTRL WD
0635 DC48 B7 5003           STA A  PIA+3       ;TURN LAMP ON
0636 DC4B 97 06             STA A  OVLD1       ;SET OVRLD COND 1 FLAG
0637 DC4D 7F 0003 OLC1A     CLR    OVRLD       ;CLR OVRLD FREQ CTRL
0638 DC50 39                RTS
0639                * NO OVERLOAD EXISTS. CLEAR BOTH OVERLOAD FLAGS
0640                *  AND TURN OFF OVERLOAD LAMP
0641                *
0642 DC51 96 04     NOLC    LDA A  INHIB          ;LD RESTART INHIBIT
0643 DC53 26 0E             BNE    NOLCC             ;SKIP PREVENT OLD CLR
0644 DC55 86 34             LDA A  #$34        ;LD OC LAMP CONTROL WD
0645 DC57 B7 5003           STA A  PIA+3       ;TURN OL INDICATOR OFF
0646 DC5A 7F 0003           CLR    OVRLD       ;CLEAR OVERLOAD FLAG
0647 DC5D 7F 0006           CLR    OVLD1       ;CLR COND 1 FLAG
0648 DC60 7F 005A           CLR    SHTDN       ;CLR SHUT DOWN FLAG
0649 DC63 39        NOLCC   RTS
```

```
SEQ  LOC  OBJ       SOURCE

0651                * OVERLOAD EXCEEDS MAX. CAUSE RAMP DOWN
0652                * SET BOTH OVERLOAD FLAGS AND TURN ON
0653                * OVERLOAD LAMP
0654                *
0655 DC64 86 3C     OLCM    LDA A  #$3C        ;SET OL LAMP CTRL WD
0656 DC66 B7 5003           STA A  PIA+3       ;TURN ON OL LAMP
0657 DC69 97 03             STA A  OVRLD       ;SET FREQ CTRL FLAG
0658 DC6B 97 06             STA A  OVLD1       ;SET COND1 FLAG
0659 DC6D 39                RTS
0660                * OUT OF CONTROL (V* > OUTCT LIMIT)
0661 DC6E 86 3C     OTCT    LDA A  #$3C        ;SET FLAG
0662 DC70 97 5A             STA A  SHTDN       ;STORE FLAG
0663 DC72 20 F0             BRA    OLCM        ;SKIP TO SHOW OVRLD
```

```
LOC  OBJ        SOURCE

                *
                * * * * * * * * * * * * * * * * * *
                * ADIN--A/D CONVERTER INPUT ROUTINE
                * * * * * * * * * * * * * * * * * *
                *
                * THIS SUBROUTINE IS USED ONLY TO START UP
                *   SUBROUTINES FUNCTIONS ARE :
                *
                *       BRING IN B+          - BPRAW
                *       BRING IN V*          - VSRAW
                *       BRING IN SPEED CMD   - FCRAW
                *
                * THESE UNIT COMMANDS AND PARAMETEPS ARE REQUIRED
                *   BY "MCL" TO DECIDE WHAT TO DO IF THE MONITOR
                *   IS NOT CONNECTED
                *
                *  START CONVERT ON B+ - THEN GET V* MS & LS
                *
                * INITIAL CONDITIONS :  START CONV = 0
                *                       B+ ADDRESS IS ENABLED
                *
           DCA0         ORG    $DCA0
                *
                * START A/D CONVERTER - UNIT IS STOPPED
                *   MUST SET PTM2 TIMER #1 TO RUN GENERATING INTERVAL
                *  MODE FOR THIS TIMER MUST NOT INTERFERE WITH STOPPED
                *   CONDITION.
                *
                * PTM2 T1 - MODE
                *   16 BIT COUNTER, SINGLE SHOT, INTERRUPT DISABLED
                *    OUTPUT ENABLED, ALLOWED TO OPERATE
                *
DCA0 86 01      ADIN    LDA A  #$01          ;SET CR2 CTRL WD
DCA2 B7 3001            STA A  PTM2+1        ;ENABLE WR TO CR1
DCA5 86 A2              LDA A  #$A2          ;SET CR1 CTRL WD
DCA7 B7 3000            STA A  PTM2          ;WR CR1 WD
DCAA CE 0060            LDX    #$0060        ;SET INTERVAL TO 96US
DCAD FF 3002            STX    PTM2+2        ;SET COUNT TO T1
                * CONTINUE WHILE A/D RUNS
DCB0 86 77              LDA A  #$77          ;LD CMD TO GET V*
DCB2 B7 5002            STA A  PIA+2         ;ENB TO GET V*
DCB5 B6 5000            LDA A  PIA           ;LD V* MS
DCB8 97 3A              STA A  VSRAW         ;SAVE V* MS
DCBA 86 75              LDA A  #$75          ;LD CMD TO GET V* LS
DCBC B7 5002            STA A  PIA+2         ;ENB FOR V* LS
DCBF B6 5000            LDA A  PIA           ;GET V* LS
DCC2 97 3B              STA A  VSRAW+1       ;SAVE V* LS
DCC4 86 76              LDA A  #$76          ;LD CMD FOR B+
DCC6 B7 5002            STA A  PIA+2         ;ENB B+
                * SET MUX ADDRESS FOR SPEED COMMAND
DCC9 C6 72              LDA B  #$72          ;MUX ADDR CHANGE
DCCB B6 5000            LDA A  PIA           ;LD B+
DCCE F7 5002            STA B  PIA+2         ;ENB SPD CMD
DCD1 97 4A              STA A  BPRAW         ;SAVE B+
```

```
SEQ  LOC  OBJ       SOURCE

0719                * BRING IN SPEED COMMAND
0720 DCD3 86 60            LDA A   #$60          ;SET CMD
0721 DCD5 B7 5002          STA A   PIA+2         ;START CONV ON SPD CMD
0722 DCD8 C6 66            LDA B   #$66          ;SET MUX ADDR CHANGE
0723 DCDA B6 5000          LDA A   PIA           ;LD SPD CMD
0724 DCDD F7 5002          STA B   PIA+2         ;ENB B+ AT MUX
0725 DCE0 97 38            STA A   FCRAW         ;SAVE SPD CMD
0726                * ANALOG MUX ADDRESS HAS BEEN LEFT AT B+ ENABLED(PB2=1)
0727                *  AND START CONVERT=0.  REMEMBER THIS SINCE START
0728                *  CONVERT IS INITIATED BEFORE ANY MORE ADDRESS CHANGES
0729                *
0730                *  HAVE ALSO LEFT PR/RST IN NO CHANGE CONDITION
0731 DCE2 39               RTS
```

```
SEQ  LOC  OBJ       SOURCE

                    *
                    * THIS SUBROUTINE ESTABLISHES FCMD AND E* FOR USE BY
                    *   THE MAIN CONTROL LOOP.  TO ACCOMPLISH THIS FUNCTION
                    *     - THE FOLLOWING PROCESSES MUST HAPPEN
                    * FCMD - GENERATION
                    *      DETERMINE IF KEYBOARD OR REAL INPUT IS USED
                    *      DOING THE CORRECT PROCESS FOR THE SOURCE OF
                    *      THIS DATA
                    *
                    * E* - GENERATION
                    *      DETERMINE IF KEYBOARD OR REAL INPUT IS USED
                    *      DOING CORRECT PROCESS FOR DATA SOURCE
                    *
                    * BOTH THESE PROCESSES REQUIRE MPA TO KNOW IF THE
                    *  KEYBOARD IS CONNECTED.
                    *
                    * IT IS KNOW THAT THE RAW DATA IA STORED AT :
                    *      BPRAW = B+ FROM A/D
                    *      FCRAW = SPEED COMMAND FROM A/D
                    *      VSRAW = V* FROM MPB
                    *
     DCE3 96 51     CSPDC   LDA A  MONFG        ;IS MONITOR CONNECTED
     DCE5 27 46             BEQ    NPROC        ;SKIP IF NO KBD AVAILABLE
     DCE7 20 17             BRA    RACIA        ;SKIP FOR KBD INPUT
                    *
                    * SECTION READS KEYBOARD TO SEE IF THERE'S ANY NEW
                    *  INSTRUCTIONS FOR THE MANUAL OPERATION OF THE UNIT
                    *
                    * AVAILABLE MANUAL CONTROLS :
                    *
                    *     "+" = ADVANCE FREQ CMD (FCMD) BY 1 HZ.
                    *     "-" = DECREMENT FREQ CMD (FCMD) BY 1 HZ.
                    *     "A" = INCREASE OUTPUT VOLTAGE(SF) BY (XXX) VOLTS
                    *     "D" = DECREASE OUTPUT VOLTAGE(SF) BY (XXX) VOLTS
                    *     "V" = USE V* FROM MPB TO OPERATE (CLOSE LOOP)
                    *     "R" = RAMP DOWN TO 0 SPEED AND REVERSE
                    *           MOTOR DIRECTION FLAG
                    *
                    *    "R" MODE CANCELED BY HITTING "+","-","A",OR "D"
                    *    "V" MODE CANCELED BY HITTING "A" OR "D"
                    *
               DD00         ORG    $DD00
                    *
     DD00 B6 6000   RACIA   LDA A  ACIAS        ;LD ACIA STATUS
     DD03 47                ASR A               ;CARRY=RDRF
     DD04 24 1D             BCC    RTRA         ;BRANCH IF NO CHANGE
     DD06 B6 6001           LDA A  ACIAD        ;LD CHAR
     DD09 84 7F             AND A  #$7F         ;STRIP PARITY
     DD0B 81 52             CMP A  #'R          ;TEST FOR REVERSE
     DD0D 27 54             BEQ    NRMO         ;BRANCH IF REVERSE
     DD0F 81 2B             CMP A  #'+          ;TEST FOR INCR FREQ
     DD11 27 5A             BEQ    IFCD         ;BRANCH TO INCR FREQ
     DD13 81 2D             CMP A  #'-          ;TEST FOR DECR FREQ
     DD15 27 5C             BEQ    DFCD         ;BRANCH TO DECR FREQ
```

```
SEQ  LOC  OBJ      SOURCE

0787 DD17 81 56             CMP A  #'V          ;TEST FOR CLOSED LOOP
0788 DD19 27 62             BEQ    UVSTAR       ;SKIP TO USE MPB-V*
0789 DD1B 81 41             CMP A  #'A          ;TEST FOR ADV V*
0790 DD1D 27 63             BEQ    AVSTAR       ;SKIP TO ADV V*
0791 DD1F 81 44             CMP A  #'D          ;TEST FOR DECR V*
0792 DD21 27 71             BEQ    DVSTAR       ;SKIP TO DECR V*
0793               * CHARACTER NOT VALID OR NO CHANGE FROM KEYBOARD
0794               *  MUST ALWAYS DO E* CALCULATION
0795               *
0796 DD23 96 E6    RTRA     LDA A  NORFG        ;LD NORM OP FLAG
0797 DD25 27 06             BEQ    NPROC        ;SKIP FOR NORM OP
0798 DD27 96 E7             LDA A  VSFLG        ;LD V* FLAG
0799 DD29 27 16             BEQ    VSPROC       ;SKIP FOR CLOSED LOOP OP
0800 DD2B 20 1D             BRA    ESC          ;COMPUTE E*
0801               *
0802               * NORMAL PROCESSING - KEYBOARD IS NOT CONNECTED
0803               *
0804               * 1ST OPERATION - DETERMINE FREQUENCY COMMAND
0805               *   THIS BRANCH MAY ALSO BE REACHED IF NORMAL MODE
0806               *  "N" IS CHOSEN FROM KEYBOARD
0807               *
0808               *
0809               *   SPEED INPUT IS DIGITAL AT PIA
0810               *   FORMAT IS  B7  B6    B5    B4  B3  B2  B1  B0
0811               *              REV UP    DN    MON  X   X   X   X
0812               *
0813               *  TOGGLE SWITCH INPUT FOR RAMPING UP SPEED, RAMPING DOWN
0814               *  SPEED, OR REVERSING MOTOR DIRECTION
0815               *
0816 DD2D 96 46    NPROC    LDA A  FCMD         ;GET COMMANDED FREQ
0817 DD2F 91 4C             CMP A  FOUT         ;COMPARE WITH ACTUAL FREQ
0818 DD31 26 0E             BNE    VSPROC       ;BRANCH IF NOT AT COMMANDED SPD
0819 DD33 D6 38             LDA B  FCRAW        ;LD FREQ CMD-TOGGLE SWITCHES
0820 DD35 C5 80             BIT B  #$80         ;TEST FOR REVERSE
0821 DD37 27 2A             BEQ    NRMO         ;IF BIT 7=0 REVERSE MOTOR
0822 DD39 C5 40             BIT B  #$40         ;TEST FOR RAMP-UP SPEED
0823 DD3B 27 2A             BEQ    IFCD1        ;IF BIT 6=0 INCREMENT SPD CMD
0824 DD3D C5 20             BIT B  #$20         ;TEST FOR RAMP-DOWN SPEED
0825 DD3F 27 32             BEQ    DFCD         ;IF BIT 5=0 RAMP-DOWN SPEED
```

```
SEQ  LOC  OBJ      SOURCE

                   *
                   * FREQ CMD DONE - NEXT IS CALCULATE E*
                   *
                   * E* = 1 + V*/B+NOM = 1 + V*/128
                   * V* FORMAT IS TWO BYTES AS :
                   * B11 B10 B9 B8 B7 B6 B5 B4 B3 B2 B1 B0 0 0 0 0
                   *                            1 . . 5 . 25 . 125
                   *
                   * WILL BE USED WITH SF CALCULATION
                   * SF SCALING :
                   * B15 B14 B13 B12 B11 B10 B9 B8 B7 B6 B5 B4 B3 B2 B1 B0
                   *  X   X   X   X   S   4   2  1. . 5  X  X  X  X  X  X 2E-5
                   *
                   * ORIGINALLY V* HAD A SIGIFICANCE(MEASURED BY OPERATION)
                   *  OF THE LSB = 0.125 VOLTS
                   *
                   * WILL SCALE V* SO THAT A "1" VOLT CHANGE AFFECTS S.F.
                   *  THE SAME AS A "1" VOLT B+ CHANGE (MEANS LS 2 OF V*
                   *  ARE OF NO IMPORTANCE SINCE THEY FALL OFF
                   *
                   * V* SCALING *--------A--------* *------B----------*
                   * VSRAW       S S S 32 16 8 4 2 1. . 5 . 25 X 0 0 0 0
                   * E*          S S S  S  8 4 2 1. . 5. 25  X X X X X X
                   *
                   * V*/128      S32 16 8  4 2 1. . 5 S S   S S S S S S
                   *
     DD41 DE 3A    VSPROC  LDX    VSRAW       ;LD V*
     DD43 DF 48            STX    VSTAR       ;STORE V* FOR CALC
     DD45 CE 0200          LDX    #$0200      ;V/F MOD 5-3-83
     DD48 20 16            BRA    VOLTHZ
     DD4A 96 48    ESC     LDA A  VSTAR       ;RELOAD MS V*
     DD4C D6 49            LDA B  VSTAR+1     ;RELOAD LS V*
     DD4E CE 0003          LDX    #$0003      ;LD SCALING CNT
     DD51 47       SVS     ASR A              ;SHIFT MS
     DD52 56               ROR B              ;SHIFT LS
     DD53 09               DEX                ;DECR SHIFT CNT
     DD54 26 FB            BNE    SVS         ;SKIP IF MORE SCALING
                   * V* IS ORIGINAL VALUE INPUT / BY 8
                   *   NOW DO 1 + V*/8
     DD56 CB 00    FESTAR  ADD B  #$00        ;ADD LS + 0.0
     DD58 89 01            ADC A  #$01        ;ADD MS + 1
     DD5A D7 41            STA B  ESRT+1      ;SAVE LS E*
     DD5C 97 40            STA A  ESRT        ;SAVE MS E*
     DD5E DE 40            LDX    ESRT        ;RELOAD NEW E*
     DD60 DF 3C    VOLTHZ  STX    ESTAR       ;STORE E* FOR INTRP PROG
     DD62 39               RTS
```

```
LOC  OBJ      SOURCE

                  *
                  * "R" - REVERSE DIRECTION KEY
                  *
DD63 73 0009 NRMO    COM     DIRCMD       ;COMPLIMENT THE DIRECTION FLAG
DD66 39              RTS                  ;RETURN
                  *
                  * "+" ADVANCE FREQ KEY
                  *
DD67 96 46   IFCD1   LDA A   FCMD         ;TEST FOR MAX FREQ
DD69 81 20           CMP A   #$20         ;MAX = 32 HZ FOR NSRDC
DD6B 27 05           BEQ     RETURN       ;IF AT MAX, RETURN
DD6D 97 E6   IFCD    STA A   NORFG        ;STORE FLAG
DD6F 7C 0046         INC     FCMD         ;INCR FCMD
DD72 39      RETURN  RTS
                  *
                  * "-" DECREMENT FREQUENCY KEY
                  *
DD73 97 E6   DFCD    STA A   NORFG        ;STORE FLAGE
DD75 96 46           LDA A   FCMD         ;LD FREQ TO BE OUTPUT
DD77 27 03           BEQ     EX1          ;BRANCH IF 0 FREQ
DD79 7A 0046         DEC     FCMD         ;DECR FREQ CMD
DD7C 39      EX1     RTS                  ;RETURN
                  *
                  * "V" USE REAL V* -CLOSE LOOP ON MPB ERROR
                  *
DD7D 7F 00E7 UVSTAR  CLR     VSFLG        ;CLR V* CLOSED LOOP FLG
DD80 20 BF           BRA     VSPROC       ;DO NORM V* PROC
                  *
                  * "A" ADVANCE V* KEY
                  *
DD82 97 E7   AVSTAR  STA A   VSFLG        ;SET KBD CTRL OF V*
DD84 97 E6           STA A   NORFG        ;RESET NORM FLAG
DD86 96 48           LDA A   VSTAR        ;V* MS
DD88 D6 49           LDA B   VSTAR+1      ;V* LS
DD8A CB 40           ADD B   #$40         ;INCR LS
DD8C 89 00           ADC A   #$00         ;INCR MS
DD8E 97 48           STA A   VSTAR        ;SAVE NEW V* MS
DD90 D7 49           STA B   VSTAR+1      ;SAVE NEW V* LS
DD92 20 B6           BRA     ESC          ;SKIP TO USE V*
                  *
                  * "D" DECREMENT V* KEY
                  *
DD94 97 E7   DVSTAR  STA A   VSFLG        ;SET KBD CTRL OF V*
DD96 97 E6           STA A   NORFG        ;RESET NORM FLAG
DD98 96 48           LDA A   VSTAR        ;V* MS
DD9A D6 49           LDA B   VSTAR+1      ;V* LS
DD9C C0 40           SUB B   #$40         ;DECR LS
DD9E 82 00           SBC A   #$00         ;DECR MS
DDA0 97 48           STA A   VSTAR        ;SAVE NEW V* MS
DDA2 D7 49           STA B   VSTAR+1      ;SAVE NEW V* LS
DDA4 20 A4           BRA     ESC          ;SKIP TO USE V*
```

```
SEQ  LOC  OBJ      SOURCE

0926                *
0927                *  STOP ROUTINE
0928                *
0929 DDA6 96 09     STROUT  LDA A  DIRCMD      ;LOAD NEW DIRECTION
0930 DDA8 97 10             STA A  DIROUT      ;STORE NEW DIRECTION
0931 DDAA 86 01             LDA A  #$01
0932 DDAC 97 01             STA A  MSTOP       ;SET UNIT STOPPED FLAG
0933 DDAE 7E D8F4           JMP    MCLCN       ;MOTOR CONNECT MCL
0934                        END

0934 LINES ASSEMBLED,  LOC = DDB1,  0000 ERRORS DETECTED.
```

SYMBOL	VALUE	ATTR	LOCN	LINK
MSTOP	0001	84	2AA2	
MSTART	0002	84	2AAB	
OVRLD	0003	84	2AB4	
INHIB	0004	84	2ABD	
PIACMD	0044	84	2AC6	
FCMD	0046	84	2ACF	
FOUT	004C	84	2AD8	
VSRAW	003A	84	2AE1	
BPRAW	004A	84	2AEA	
FCRAW	0038	84	2AF3	
DOMID	0020	84	2AFC	
DPF	0022	84	2B05	
J	0024	84	2B0E	
F60DEG	00C0	84	2B17	
FTINCR	00C8	84	2B20	
MONFG	0051	84	2B29	
NPPCT	00C5	84	2B32	
NORFG	00E6	84	2B3B	
VSFLG	00E7	84	2B44	
PTIS	E000	84	2B4D	
FTPNW	00C1	84	2B56	
OVLD1	0006	84	2B5F	
INDEX	00E8	84	2B68	
FTPTR	00C3	84	2B71	
ACIAC	6000	84	2B7A	
DIRCMD	0009	84	2B83	
DIROUT	0010	84	2B8C	
BPFLG	0050	84	2B95	
VSTAR	0048	84	2B9E	
ESOC	00EA	84	2BA7	
K	0005	84	2BB0	
AVE	00F0	84	2BB9	
ESMM	0001	84	2BC2	
ESML	0000	84	2BCB	
ES1MM	0000	84	2BD4	
ES1ML	0080	84	2BDD	
ESTAR	003C	84	2BE6	
YN	00F0	84	2BEF	
FTPNT	00CA	84	2BF8	
ESRT	0040	84	2C01	
OVCMP	00F0	84	2C0A	
VPADDR	00D8	84	2C13	
VSREF	00DB	84	2C1C	
OUTCTL	00EF	84	2C25	
OUTCTM	000D	84	2C2E	
SHTDN	005A	84	2C37	
PIA	5000	84	2C40	
PTM1	4000	84	2C49	
PTM2	3000	84	2C52	
WLATCH	2007	84	2C5B	
ACIAS	6000	84	2C64	
ACIAD	6001	84	2C6D	
INIT	D800	84	2C76	
INIT1	D80F	84	2C7F	

SYMBOL	VALUE	ATTR	LOCN	LINK
TRESET	DA60	84	2C88	
MCL	D880	84	2C91	
MCL02	D89B	84	2C9A	
ADIN	DCA0	84	2CA3	
CSPDC	DCE3	84	2CAC	
MCL01	D893	84	2CB5	
PROTCT	DC00	84	2CBE	
MCL03	D8B7	84	2CC7	
OLRMPD	D8FF	84	2CD0	
TFOUT	D8CF	84	2CD9	
MCL04	D8C8	84	2CE2	
RAMPDN	DB80	84	2CEB	
MCL3A	D8BA	84	2CF4	
RAMPUP	DB00	84	2CFD	
STUNT	D8F6	84	2D06	
CHPTR	D8D3	84	2D0F	
AINDX	D8DB	84	2D18	
TSTART	DA20	84	2D21	
MCLCN	D8F4	84	2D2A	
STROUT	DDA6	84	2D33	
OLRMP1	D908	84	2D3C	
SLWRD	DB91	84	2D45	
SMOUT	DB11	84	2D4E	
SFOUT	DB0C	84	2D57	
ERU	DB10	84	2D60	
ERD	DB8C	84	2D69	
ESRD	DBAA	84	2D72	
PVOLT	DC0F	84	2D7B	
ESPRT	DC13	84	2D84	
OTCT	DC6E	84	2D8D	
NOLC	DC51	84	2D96	
PRTCT1	DC32	84	2D9F	
OLC1	DC46	84	2DA8	
OLCM	DC64	84	2DB1	
OLC1A	DC4D	84	2DBA	
NOLCC	DC63	84	2DC3	
NPROC	DD2D	84	2DCC	
RACIA	DD00	84	2DD5	
RTRA	DD23	84	2DDE	
NRMO	DD63	84	2DE7	
IFCD	DD6D	84	2DF0	
DFCD	DD73	84	2DF9	
UVSTAR	DD7D	84	2E02	
AVSTAR	DD82	84	2E0B	
DVSTAR	DD94	84	2E14	
VSPROC	DD41	84	2E1D	
ESC	DD4A	84	2E26	
IFCD1	DD67	84	2E2F	
VOLTHZ	DD60	84	2E38	
SVS	DD51	84	2E41	
FESTAR	DD56	84	2E4A	
RETURN	DD72	84	2E53	
EX1	DD7C	C4	2E5C	

END ASM V1.1

```
LOC  OBJ  SOURCE

                * PTM INTERRUPT SERVICE ROUTINE
                        NAM     PTMS6
                ***************************************
                * PTM-PULSE TIMER INTERRUPT ROUTINE *
                ***************************************
                * THIS MODULE CONTROLS THE OPERATION OF 2 PULSE TIMER MODULES
                * IN ORDER TO SYNTHESIZE THE PULSE WIDTH MODULATED (PWM)
                * WAVEFORMS REQUIRED TO CONTROL A UNIT.  PTM#1 OUTPUTS ARE
                * CONNECTED TO THE THREE PHASE INPUTS OF THE CONTROLLER.
                * PTM#2 ESTABLISHES THE PULSE PERIOD INTERVAL AND GENERATES
                * INTERRUPTS WHICH USE THIS MODULE.
                *
                *
                * DEFINITION OF MEMORY LOCATIONS USED BY THIS ROUTINE
                *
                *   (LOCAL)
     00C0 F60DG   EQU     $00C0            ;START OF 60 DEG SEGMENT FLAG
     00C1 FTPNW   EQU     $00C1            ;NEW FREQ TABLE POINTER
     00C3 FTPTR   EQU     $00C3            ;POINTER FOR VALUES IN USE
     00C6 NPPMD   EQU     $00C6            ;COUNT FOR MID-POINT OF P.P.
     00D2 BSTAR   EQU     $00D2            ;B* FOR SF COMPUTATION
     00D0 SF      EQU     $00D0            ;SCALE FACTOR
     0072 PWDOM   EQU     $0072            ;DOMINANT@S PULSE WIDTH
     0074 PWCOM   EQU     $0074            ;COMPLEMENTS@S PULSE WIDTH
     0070 TPP     EQU     $0070            ;WIDTH OF PULSE PERIOD
     0078 T1      EQU     $0078            ;1ST SWITCHING TIME
     007A T2      EQU     $007A            ;2ND SWITCHING TIME
     007C T3      EQU     $007C            ;3RD SWITCHING TIME
     00D8 VPADD   EQU     $00D8            ;V PEAK TABLE ADDRESS
                *
                *
     0030 SFMAX   EQU     $0030            ;MAX SF FOR PP
     0032 SFMIN   EQU     $0032            ;MIN SF FOR PP
                *
                *
                *   (GLOBAL)
     0044 PIACMD  EQU     $0044            ;COMMANDED STATE OF PIA-B
     0020 DOMID   EQU     $0020            ;DOMINANT'S ID (0=A,1=B,2=C)
     0022 DPF     EQU     $0022            ;DOM'S POLARITY (0=+ ,FF=-)
     0024 J       EQU     $0024            ;FLAG (+=1ST,ODD/-=1ST EVEN)
     00C8 FTINCR  EQU     $00C8            ;FREQ TAB (+4=1ST HALF/-4=2ND)
     0051 MONFG   EQU     $0051            ;MON CONN FLAG ( 0= NOT CONN)
     003A VSRAW   EQU     $003A            ;V* FROM PIA (RAW DATA)
     00E7 VSFLG   EQU     $00E7            ;V* IN FLG(0=IN MUPA,1=USE KBD)
     004A BFRAW   EQU     $004A            ;B+ (RAW INPUT)
     0038 FCRAW   EQU     $0038            ;FREQ CMD FROM A/D (RAW DATA)
     003C ESTAR   EQU     $003C            ;E* = V*/B+NOM
     00C5 NPPCT   EQU     $00C5            ;NO.  OF P.P.'S LEFT IN SEG
     004C FOUT    EQU     $004C            ;OUTPUT FREQUENCY
     0046 FCMD    EQU     $0046
     0009 DIRCMD  EQU     $0009
     0010 DIROUT  EQU     $0010
```

```
SEQ  LOC  OBJ      SOURCE

                   *
                   *
                   *  ( I/O ADDRESSES)
          2000 MPIER    EQU    $2000          ;MULTPLR 12 BITS (HANDLE AS 1
          2002 MCAND    EQU    $2002          ;MULTPLCND 12 BITS ( "    "
          2005 PROD     EQU    $2005          ;PRODUCT (MS 8 + 2 MORE BYTES)
          5000 PIA      EQU    $5000          ;PROGRAMMABLE INTERFACE ADAPT
          4000 PTM1     EQU    $4000          ;PULSE TIMER MODULE #1
          3000 PTM2     EQU    $3000          ;PULSE TIMER MODULE #2
          2007 WLATCH   EQU    $2007          ; 'W' LATCH ADDRESS (FOR MUPB)
                   *
                   *
                   *
                   *
          0001          LSYMB  1
          E000          ORG    $E000
                   *INTERRUPT SERVICE
                   *
                   * IMMEDIATELY CLEAR INTERRUPT FLAG IN TIMER
                   *    THIS DONE VIA FOLLOWING ADDRESSING IF TIMER #2
     E000 B6 3001 PTIS     LDA A  PTM2+1        ;READ TIMER STATUS REG
     E003 B6 3002          LDA A  PTM2+2        ;READ TIMER #1 COUNT
                   *
                   *CHECK IF THIS IS 60 DEG SEGMENT BOUNDARY
                   *
     E006 7D 0000          TST    F60DG          ;TEST FOR BEGINNING OF 60 DE
     E009 27 4A            BEQ    NOT60          ;BRANCH IF NOT BEGINNING
                   * 60 DEG INTERVAL - SET UP COUNTERS, FLIP-FLOPS, ETC
                   *
                   *  FIRST THING IS SET MUX ADDRESS AND START CONVERT TO
                   *                         SPEED CMD        "0"
                   *
     E00B 86 66            LDA A  #$66          ;ENAB B+ & SET SC=0
     E00D B7 5002          STA A  PIA+2         ;START CONV TO CLR SEQ
                   * THE ABOVE A/D CONVERSION IS USED ONLY TO GET START CONVERT
                   *   BACK IN SEQUENCE. CONVERTER RESULT IS NOT USED !!
                   *
     E010 DE C1            LDX    FTPNW          ;GET NEW FREQ TABLE POINTER
     E012 DF C3            STX    FTPTR          ;SET POINTER TO USE NEW VALU
     E014 A6 00            LDA A  00,X          ;LOAD NO. OF P.P.'S IN SEG
     E016 97 C5            STA A  NPPCT         ;ESTABLISH BEGIN. PP COUNT
                   *
                   * CALCULATE MID-POINT OF SEGMENT
                   *
     E018 4C               INC A                ;NPP+1
     E019 47               ASR A                ;(NPP+1)/2=MID-POINT COUNT
     E01A 97 C6            STA A  NPPMD         ;STORE IN MID-POINT LOCAL
                   *
                   * PULL PULSE PERIOD PARAMETERS FROM TABLE
                   *
     E01C EE 01            LDX    01,X           ;GET PP LENGTH
     E01E FF 3002          STX    PTM2+2         ;ESTAB TIME BETWEEN INTERRU
     E021 DF 70            STX    TPP            ;STORE NEW PP LENGTH
     E023 DE C3            LDX    FTPTR          ;RELOAD TABLE TOP ADDRESS
```

```
SEQ  LOC  OBJ      SOURCE

     E025 EE 05            LDX    05,X           ;LOAD MAX SF
     E027 DF 30            STX    SFMAX          ;STORE AS MAX SCALE FACTOR
     E029 DE C3            LDX    FTPTR          ;RELOAD TABLE TOP
     E02B EE 07            LDX    07,X           ;LOAD MIN SF
     E02D DF 32            STX    SFMIN          ;STORE AS MIN SCALE FACTOR
                   *
                   * UPDATE DOMINANT WAVEFORM ID
                   * DOMINANT ID = DOMID = (A=0,B=1,C=2)
                   *
     E02F 96 20            LDA A  DOMID        ;LOAD DOM ID FLAG
                   *
                   * ID ALWAYS CHANGES B,A,C,B,A,C, ETC
                   * AT 60 DEG BOUNDARY DOMID=B !
                   *
     E031 4A               DEC A             ;DECREMENT ID
     E032 2A 02            BPL    AA             ;BRANCH IF AOR B
     E034 86 02            LDA A  #$02         ;SET FLAG VALUE FOR "C"
     E036 97 20    AA      STA A  DOMID        ;STORE NEW ID
```

```
SEQ  LOC  OBJ      SOURCE

                   * CHANGE DOMINANT POLARITY FLAG (DPF=0/+ , FF/-)
                   * ALSO SET VALUES OF A,B,C FLIPFLOPS.  AT 60 DEG BOUNDARY DPF=0
                   *  LAST START CONVERT ADDR.-PB2=0 BEFORE HERE
                   *
     E038 86 A6           LDA A  #$A6            ;LD CMD TO SET FLIP FLOP
     E03A 73 0022         COM    DPF               ;CHANGE DOMINANT POLARITY
     E03D 2B 02           BMI    A2                ;BRANCH IF NEGATIVE
     E03F 86 C6           LDA A  #$C6            ;LD CMD TO RESET FLIP FLOP
     E041 97 44    A2     STA A  PIACMD          ;STORE COMMAND FOR LATER USE
                   *
                   * INITIALIZE INDICES FOR PROCESSING
                   *
     E043 7F 0024         CLR    J                 ;CLEAR 1ST/2ND HALF FLAG
     E046 CE 0004         LDX    #+4             ;SET 1ST HALF VALUE
     E049 DF C8           STX    FTINCR          ;ESTABLISH LS OF ADDRESS
                   *
                   * SEND OUTPUT FREQUENCY TO MUPB
                   *
     E04B 96 4C           LDA A  FOUT            ;LOAD PRESENT OUTPUT FREQUENCY
     E04D B7 2007         STA A  WLATCH          ;STORE FOUT TO LATCH
     E050 7F 00C0         CLR    F60DG             ;CLEAR 60 DEG BOUNDARY FLAG
     E053 20 28           BRA    BPIN              ;DONE 60 DEG SEG, DO B+ INPUT
                   *
                   * CODE IF NOT A 60 DEG BOUNDARY (INSIDE A 60 DEG SEGMENT)
                   * USE EXTRA TIME TO INPUT V* AND FCMD
                   * SPD CMD AND V* ARE INPUT AND STORED FOR USE BY MAIN PROGRAM
                   * A/D START CONVERT IS HIGH TO BEGIN WITH
                   *    ADDRESS FOR SPD CMD ALREADY SET
                   *
     E055 86 62    NO160  LDA A  #$62            ;SET IN RESET/PRESET - NO CHAN
     E057 C6 63           LDA B  #$63            ;SET FOR V* INPUT
     E059 B7 5002         STA A  PIA+2           ;START A/D CONVERSION
                   *
                   * CONVERSION FOR SPEED COMMAND IS STARTED
                   * GET V* WHILE CONVERSION IS WORKING
                   * START CONVERSION - PIA-PB4 SET=1
                   *
     E05C F7 5002         STA B  PIA+2           ;OUTPUT COMMAND TO PIA (ENB V*
     E05F F6 5000         LDA B  PIA             ;BRING IN V* MSB
     E062 86 61           LDA A  #$61            ;SELECT V* LS,OTHERS NO CHANGE
     E064 B7 5002         STA A  PIA+2           ;OUTPUT COMMAND TO PIA
     E067 B6 5000         LDA A  PIA             ;LOAD V* LS
     E06A 97 3B           STA A  VSRAW+1         ;SAVE V* LS
     E06C D7 3A           STA B  VSRAW           ;SAVE V* MS
                   *  ENABLE ANALOG MUX FOR B+ INPUT
     E06E 86 60           LDA A  #$60            ;SET ENAB WD FOR B+
     E070 B7 5002         STA A  PIA+2           ;ENAB B+
                   *
                   * FINISH A/D CONVERSION FOR SPEED COMMAND
                   *
     E073 01              NOP                    ;FILL KEEP PROG LENGTH
     E074 01              NOP                    ;FILL KEEP PROG LENGTH
     E075 01              NOP                    ;FILL KEEP PROG LENGTH
     E076 01              NOP                    ;FILL KEEP PROG LENGTH
```

```
SEQ  LOC  OBJ        SOURCE

0181 E077 01                    NOP                 ;FILL KEEP PROG LENGTH
0182 E078 F6 5000               LDA B   PIA          ;LOAD SPEED COMMAND
0183 E07B D7 38                 STA B   FCRAW        ;STORE RAW FREQUENCY COMMAND
```

```
SEQ  LOC  OBJ       SOURCE

0185                *
0186                * V+ IS STORED - EITHER BY ABOVE OR VIA KEYBOARD CONTROL
0187                * CONTINUE SEGMENT PROCESSING
0188                *
0189                * BRING IN V PEAK DURING EVERY SEGMENT, CALCULATE NEW
0190                * SCALE FACTOR AND APPLY TO ADJUST PULSE WIDTHS.
0191                *
0192                * V PEAK MUST BE READ IN AND SAVED IN BPRAW.  ALSO SET AND
0193                * RESET FF'S DEPENDING ON POLE VOLTAGE STATE FOR 60 DEG
0194                * SEGMENT
0195                * INPUT V PEAK FIRST
0196                * START CONVERSION  PIA-PB4 SET=0
0197                *
0198 E07D 86 76     BPIN     LDA A   #$76          ;SAVE SET/RST,ENB A/D,SET S.C.
0199 E07F B7 5002            STA A   PIA+2         ;ENABLE FLIPFLOPS, START CONVE
0200 E082 C6 06              LDA B   #$06          ;LD A/D TIMING CNT(36 US TOTAL)
0201 E084 5A        CBPC     DEC B               ;DECREMENT A/D COUNT
0202 E085 26 FD              BNE     CBPC              ;BRANCH WAIT FOR CONVERTER
0203 E087 B6 5000            LDA A   PIA           ;LOAD B+
0204 E08A 97 4A              STA A   BPRAW         ;STORE B+(VOLT-SEC) (RAW INPUT
0205 E08C C6 72              LDA B   #$72          ;SET SPD CMD ENAB WD
0206 E08E F7 5002            STA B   PIA+2         ;ENAB SPD CMD
0207                *
0208                * V PEAK HAS NOW STORED AT BPRAW
0209                * , PIA IS SET UP SO THAT FLIPFLOPS
0210                * WILL BE SET, RESET OR NOT AFFECTED BY PULSE PERIOD INTERRUP
0211                *
0212                * NEXT STEP COMPUTES SCALE FACTOR BASED ON NEW V PEAK
0213                *
0214                * MUST FIND VPEAK NOM/V PEAK FROM LOOK UP TABLE
0215                *     FIRST GENERATE TABLE ADDRESS FROM V PEAK
0216 E091 86 E5              LDA A   #$E5          ;LD MS OF TABLE ADDR
0217 E093 D6 4A              LDA B   BPRAW         ;LD V PK FROM MEM
0218 E095 58                 ASL B               ;ADJ LS FOR 2 BYTES/V PK
0219 E096 24 02              BCC     SVPADD            ;SKIP FOR <128 SAVE ADDR
0220 E098 86 E6              LDA A   #$E6          ;SET MS AT 128 VOLT POS
0221                * ADDRESS FOUND,SAVE IT AND LOOK UP VPEAK NOM/V PEAK
0222 E09A 97 D8     SVPADD   STA A   VPADD         ;SAVE MS
0223 E09C D7 D9              STA B   VPADD+1       ;SAVE LS
0224 E09E DE D8              LDX     VPADD             ;RELOAD VPEAK ADDR
0225                *
0226                * SCALE FACTOR COMPUTATION
0227                *
0228                *  SF = (1+E*/128) * VPEAK NOM/V PEAK
0229                *         FROM MCL    FOUND BY THIS SUBROUTINE
0230                *
0231 E0A0 EE 00     SFCAL    LDX     0,X               ;LD X V PEAK NOM/V PEAK
0232 E0A2 FF 2000            STX     MPIER             ;STORE AS MULTIPLIER
0233 E0A5 DE 3C              LDX     ESTAR             ;LD (1+E*/128)
0234 E0A7 FF 2002            STX     MCAND             ;STORE AS MULTIPLICAND
0235 E0AA B6 2005            LDA A   PROD            ;LD MS OF SF
0236 E0AD F6 2006            LDA B   PROD+1          ;LD LS OF SF
0237 E0B0 47                 ASR A               ;ADJ FOR MULTIP TRUNC
0238 E0B1 56                 ROR B               ;ADJ LS FOR TRUNC
```

```
SEQ  LOC  OBJ      SOURCE

0239                 *
0240                 * SCALE FACTOR NOW EXISTS IN A & B REGISTERS
0241                 *   SCALING IS :
0242                 *  B15 B14 B13 B12 B11 B10 B9 B8 B7 B6 B5 B4 B3 B2 B1 B0
0243                 *   X   X   X   X   S   4   2  1..5  D  D  D  D  D  D 2E-8
0244                 *
0245 E0B2 D7 D1            STA B  SF+1          ;SAVE SF LS
0246 E0B4 97 D0            STA A  SF            ;SAVE SF MS
0247                 * TEST FOR NEGATIVE FIRST
0248 E0B6 81 00            CMP A  #$00          ;CMP TO ZERO
0249 E0B8 2B 12            BMI    SFMNL          ;SKIP TO LD MIN SF
```

```
SEQ  LOC  OBJ      SOURCE

                   *
                   *
                   * TEST MS OF SCALE FACTOR
                   * SCALE FACTOR NOMINAL  IS 0.5     (MIN E* & MIN B+ERR/B+NOM
                   *                           2.996  (MAX E* & MAX B+ERR/B+NOM)
                   *
     E0BA 27 0C             BEQ    SFMSZ              ;BRANCH IF MS=0
                   *
                   * SF>=1.0000
                   *
     E0BC D0 31             SUB B  SFMAX+1           ;SF-SFMAX LS
     E0BE 92 30             SBC A  SFMAX             ;SF-SFMAX MS
     E0C0 2B 0E             BMI    SFOK               ;BRANCH IF <3.00
     E0C2 DE 30             LDX    SFMAX              ;LOAD MAX SF
     E0C4 DF D0             STX    SF                 ;SET SF = SFMAX IN MEM
     E0C6 20 08             BRA    SFOK               ;BRANCH TO SFOK
                   *
                   * SF<1.000
                   *
     E0C8 D1 33    SFMSZ    CMP B  SFMIN+1           ;SF-SFMIN LS
     E0CA 22 04             BHI    SFOK               ;BRANCH IF >0.75
     E0CC DE 32    SFMNL    LDX    SFMIN              ;LOAD MIN SF
     E0CE DF D0             STX    SF                 ;SET SF = SFMIN
                   *
                   * NEXT STEP IS TO COMPUTE PULSE WIDTHS BASED ON WHERE YOU ARE
                   * COMPUTE DOMINANT FIRST
                   *
     E0D0 DE C3    SFOK     LDX    FTPTR              ;LOAD WORK FREQ TABLE PT
     E0D2 EE 09             LDX    09,X               ;GET PWDOM BEFORE SCALING
     E0D4 FF 2000           STX    MPIER              ;STORE PWDOM AS MULTIPLIER
     E0D7 DE D0             LDX    SF                 ;LOAD SCALE FACTOR
     E0D9 FF 2002           STX    MCAND              ;STORE SF AS MULTIPLICAND
                   *
                   * PWDOM = SF * TC1 (PWDOM BEFORE SCALING)
                   *
     E0DC B6 2005           LDA A  PROD              ;INPUT SIGN + PR1 TO PR7
     E0DF F6 2006           LDA B  PROD+1            ;INPUT PR8 TO PR15
     E0E2 47                ASR A                   ;POSITION ANSWER MS
     E0E3 56                ROR B                   ;POSITION LS
                   *
                   * PWDOM NOW HAS RESOLUTION IN MICROSECONDS AS
                   *
                   * B15 B14 B13 B12 B11 B10 B9 B8  B7  B6 B5 B4 B3 B2 B1 B0
                   *  S   S  8K  4K  2K  1K 512 256 128 64 32 16  8  4  2  1*
                   *
     E0E4 97 72             STA A  PWDOM             ;STORE MS
     E0E6 D7 73             STA B  PWDOM+1           ;STORE LS
```

```
SEQ  LOC  OBJ        SOURCE

0299                 *
0300                 * NEXT STEP IS TO COMPUTE THE COMPLEMENT PULSE WIDTH
0301                 *
0302 E0E8 DE C3             LDX    FTPTR             ; RELOAD THE TABLE POINTER
0303 E0EA EE 0B             LDX    11,X              ;GET TC - COMPLEMENT
0304 E0EC FF 2000           STX    MPIER             ;STORE AS MULTIPLIER
0305                 *
0306                 * DON'T HAVE TO RELOAD SCALE FACTOR SINCE IT IS LATCHED AS
0307                 *  MULTIPLICAND
0308                 *
0309 E0EF B6 2005           LDA A  PROD             ;LOAD MS 8
0310 E0F2 F6 2006           LDA B  PROD+1           ;LOAD LS 8
0311 E0F5 47                ASR A                 ;POSITION MS
0312 E0F6 56                ROR B                 ;POSITION LS
0313 E0F7 97 74             STA A  PWCOM            ;STORE MS
0314 E0F9 D7 75             STA B  PWCOM+1          ;STORE LS
0315                 *
0316                 * RESOLUTION & SCALING OF PWCOM IS IDENTICAL TO PWDOM
```

```
SEQ  LOC  OBJ       SOURCE

                    *
                    * CALCULATE THE THREE SWITCHING TIMES FOR THIS PULSE PERIOD
                    * TEST THE 'J' INDEX TO DETERMINE WHICH OF TWO ALGORITHMS TO
                    * USE DIFFERENCE IN ALGORITHM IS RESULT OF POSITION IN SEGME
                    * AND WHETHER PP NO. IS ODD OR EVEN
                    *
                    * T1= 1ST TO TIME OUT
                    * T2= 2ND TO TIME OUT
                    * T3= 3RD TO TIME OUT
                    *                                     ODD P.P'S      EVEN P.P.
                    * IN 1ST HALF OF SEGMENT     T3=    T3-PWCOM        T1+PWCOM
                    * IN 2ND HALF OF SEGMENT     T3=    T1+PWCOM        T3-PWCOM
                    * J=(-) IF 1ST HALF EVEN PP, OR IF 2ND HALF ODD  PP
                    * J=(+) IF 1ST HALF ODD  PP, OR IF 2ND HALF EVEN PP
                    *
     E0FB 7D 0024          TST    J                  ;IS J +/- ?
     E0FE 2B 20            BMI    MINDX              ;BRANCH FOR (-) INDEX
                    *
                    * ALGORITHM 1
                    * CALCULATE SWITCHING TIMES FOR:
                    *     PVS#DPF AND PWCOM IS 2ND COMPLEMENT
                    *  OR PVS=DPF AND PWCOM IS 1ST COMPLEMENT
                    *
                    * T1=(TPP-PWDOM)/2
                    *
     E100 96 70            LDA A  TPP               ;LOAD TOTAL PP COUNT MS
     E102 D6 71            LDA B  TPP+1             ;LOAD    "     "    "   LS
     E104 D0 73            SUB B  PWDOM+1           ;TPP-PWDOM LS
     E106 92 72            SBC A  PWDOM             ;TPP-PWDOM MS
     E108 47               ASR A                  ;DIVIDE DIFFERENCE
     E109 56               ROR B                  ;  BY 2
     E10A 97 78            STA A  T1                ;STORE MS
     E10C D7 79            STA B  T1+1              ;STORE LS
                    *
                    * T3=T1+PWDOM
                    *
     E10E DB 73            ADD B  PWDOM+1           ;T1+PWDOM LS
     E110 99 72            ADC A  PWDOM             ;T1+PWDOM MS
     E112 97 7C            STA A  T3                ;STORE MS
     E114 D7 7D            STA B  T3+1              ;STORE LS
                    *
                    * T2=T3-PWCOM
                    *
     E116 D0 75            SUB B  PWCOM+1           ;T3-PWCOM LS
     E118 92 74            SBC A  PWCOM             ;T3-PWCOM MS
     E11A 97 7A            STA A  T2                ;STORE T2 MS
     E11C D7 7B            STA B  T2+1              ;STORE T2 LS
     E11E 20 22            BRA    OTCNTS             ;BRANCH TO OUTPUT COUNTS
```

```
SEQ  LOC  OBJ      SOURCE

0367               *
0368               * CALCULATE THE SWITCHING TIMES FOR:
0369               *     PVS#DPF AND PWCOM IS THE 1ST COMP
0370               *  OR PVS=DPF AND PWCOM IS THE 2ND COMP
0371               *
0372               * T1=(TPP-PWDOM)/2
0373               *
0374 E120 96 70    MINDX   LDA A  TPP           ;LOAD TOTAL PP MS
0375 E122 D6 71            LDA B  TPP+1         ;LOAD   "     " LS
0376 E124 D0 73            SUB B  PWDOM+1       ;TPP-PWDOM LS
0377 E126 92 72            SBC A  PWDOM         ;TPP-PWDOM MS
0378 E128 47               ASR A               ;DIVIDE DIFFERENCE
0379 E129 56               ROR B               ;  BY 2
0380 E12A 97 78            STA A  T1            ;STORE MS
0381 E12C D7 79            STA B  T1+1          ;STORE LS
0382               *
0383               * T3=T1+PWDOM
0384               *
0385 E12E DB 73            ADD B  PWDOM+1       ;T1+PWDOM LS
0386 E130 99 72            ADC A  PWDOM         ;T1+PWDOM MS
0387 E132 97 7C            STA A  T3            ;STORE MS
0388 E134 D7 7D            STA B  T3+1          ;STORE LS
0389               *
0390               * T2=T1+PWCOM
0391               *
0392 E136 96 78            LDA A  T1            ;LOAD T1 MS
0393 E138 D6 79            LDA B  T1+1          ;LOAD T1 LS
0394 E13A DB 75            ADD B  PWCOM+1       ;T1+PWCOM LS
0395 E13C 99 74            ADC A  PWCOM         ;T1+PWCOM MS
0396 E13E 97 7A            STA A  T2            ;STORE MS
0397 E140 D7 7B            STA B  T2+1          ;STORE LS
0398 E142 96 09    OTCNTS  LDA A  $09           ;GET DIR CMD FLAG
0399 E144 91 10            CMP A  DIROUT        ;COMPARE WITH DIR OUT
0400 E146 27 0E            BEQ    DAN1
0401 E148 96 4C            LDA A  FOUT          ;GET OUTPUT FREQ
0402 E14A 27 0A            BEQ    DAN1
0403 E14C 86 00            LDA A  #$00
0404 E14E 97 46            STA A  FCMD          ;SET FREQ CMD TO ZERO
0405 E150 91 10            CMP A  DIROUT        ;IS DIRECTION CW?
0406 E152 27 0A            BEQ    DAN2
0407 E154 20 0B            BRA    DAN3
0408 E156 96 09    DAN1    LDA A  DIRCMD
0409 E158 97 10            STA A  DIROUT
0410 E15A 96 09            LDA A  DIRCMD
0411 E15C 26 03            BNE    DAN3
0412 E15E 7E E242  DAN2    JMP    CWRT
0413 E161 7E E342  DAN3    JMP    CCWRT
```

```
SEQ  LOC  OBJ       SOURCE

               E242         ORG     $E242
                        *
                        * CLOCKWISE ROUTINE
                        *
     E242 96 C5     CWRT    LDA A   NPPCT          ;GET NO. OF PP COUNTED
     E244 84 01             AND A   #1             ;IS IT ODD?
     E246 27 3A             BEQ     EVPP1           ;BRANCH FOR EVEN PP
                        *
                        * DOMINANT POLARITY DOES NOT EQUAL POLE VOLTAGE STATE
                        * DETERMINE THE DOMINANT PHASE (INIT: 0=A,1=B,2=C )
                        *
     E248 96 20             LDA A   DOMID          ;LOAD DOMINANT PHASE ID
     E24A 4A                DEC A                 ;DECREMENT PHASE (B-A-C ALWAYS)
     E24B 27 13             BEQ     PHBD01          ;BRANCH B PHASE DOMINANT (OD
     E24D 2A 22             BPL     PHCD01          ;BRANCH C PHASE DOMINANT (ODD
                        *
                        * PHASE A IS DOMINANT
                        * DOMINANT POLARITY DOES NOT EQUAL POLE VOLTAGE STATE A-C-B
                        *
     E24F DE 78             LDX     T1              ;LOAD T1 COUNT
     E251 FF 4002           STX     PTM1+2          ;T1 TO PHASE A (DOM/1ST ON)
     E254 DE 7A             LDX     T2              ;LOAD T2 COUNT
     E256 FF 4006           STX     PTM1+6          ;T2 TO PHASE C(2ND ON/1ST OFF
     E259 DE 7C             LDX     T3              ;LOAD T3 COUNT
     E25B FF 4004           STX     PTM1+4          ;T3 TO PHASE B (DOM/2ND OFF)
     E25E 20 5A             BRA     TMDPT1          ;BRANCH TO TEST FOR MID-POINT
                        *
                        * PHASE B IS DOMINANT  B-A-C
                        *
     E260 DE 78     PHBD01  LDX     T1              ;LOAD T1 COUNT
     E262 FF 4004           STX     PTM1+4          ;T1 TO PHASE B (DOM/1ST ON)
     E265 DE 7A             LDX     T2              ;LOAD T2 COUNT
     E267 FF 4002           STX     PTM1+2          ;T2 TO PHASE A(2ND ON/1ST OFF
     E26A DE 7C             LDX     T3              ;LOAD T3 COUNT
     E26C FF 4006           STX     PTM1+6          ;T3 TO PHASE C (DOM/2ND OFF)
     E26F 20 49             BRA     TMDPT1          ;BRANCH TO TEST FOR MID-POINT
                        *
                        * PHASE C IS DOMINANT    C-B-A
                        *
     E271 DE 78     PHCD01  LDX     T1              ;LOAD T1 COUNT
     E273 FF 4006           STX     PTM1+6          ;T1 TO PHASE C (DOM/1ST ON)
     E276 DE 7A             LDX     T2              ;LOAD T2 COUNT
     E278 FF 4004           STX     PTM1+4          ;T2 TO PHASE B(2ND ON/1ST OFF)
     E27B DE 7C             LDX     T3              ;LOAD T3 COUNT
     E27D FF 4002           STX     PTM1+2          ;T3 TO PHASE A (DOM/2ND OFF)
     E280 20 38             BRA     TMDPT1          ;BRANCH TO TEST FOR MID-POINT
```

```
SEQ  LOC  OBJ      SOURCE

0462                *
0463                * DOMINANT POLARITY EQUALS POLE VOLTAGE STATE
0464                * DETERMINE THE DOMINANT PHASE
0465                *
0466 E282 96 20     EVPP1   LDA A  DOMID          ;LOAD DOMINANT ID
0467 E284 4A                DEC A                ;DECREMENT PHASE ( -=A, 0=B, +=C)
0468 E285 27 13             BEQ    PHBDE1         ;PHASE B DOMINANT - EVEN PP
0469 E287 2A 22             BPL    PHCDE1         ;PHASE C DOMINANT - EVEN PP
0470                *
0471                * PHASE A IS DOMINANT   B-C-A
0472 E289 DE 78             LDX    T1             ;LOAD T1 COUNT
0473 E28B FF 4004           STX    PTM1+4         ;T1 TO PHASE B (DOM/2ND ON)
0474 E28E DE 7A             LDX    T2             ;LOAD T2 COUNT
0475 E290 FF 4006           STX    PTM1+6         ;T2 TO PHASE C
0476 E293 DE 7C             LDX    T3             ;LOAD T3 COUNT
0477 E295 FF 4002           STX    PTM1+2         ;T3 TO PHASE A
0478 E298 20 20             BRA    TMDPT1         ;BRANCH TO TEST FOR MID-POINT
0479                *
0480                * PHASE B IS DOMINANT C-A-B
0481                *
0482 E29A DE 78     PHBDE1  LDX    T1             ;LOAD T1 COUNT
0483 E29C FF 4006           STX    PTM1+6         ;T1 TO PHASE C
0484 E29F DE 7A             LDX    T2             ;LOAD T2 COUNT
0485 E2A1 FF 4002           STX    PTM1+2         ;T2 TO PHASE A
0486 E2A4 DE 7C             LDX    T3             ;LOAD T3 COUNT
0487 E2A6 FF 4004           STX    PTM1+4         ;T3 TO PHASE B
0488 E2A9 20 0F             BRA    TMDPT1         ;BRANCH TOTEST FOR MID-POINT
0489                *
0490                * PHASE C IS DOMINANT A-B-C
0491                *
0492 E2AB DE 78     PHCDE1  LDX    T1             ;LOAD T1 COUNT
0493 E2AD FF 4002           STX    PTM1+2         ;T1 TO PHASE A
0494 E2B0 DE 7A             LDX    T2             ;LOAD T2 COUNT
0495 E2B2 FF 4004           STX    PTM1+4         ;T2 TO PHASE B
0496 E2B5 DE 7C             LDX    T3             ;LOAD T3 COUNT
0497 E2B7 FF 4006           STX    PTM1+6         ;T3 TO PHASE C
```

```
SEQ  LOC  OBJ        SOURCE

                     *
                     * TEST IF THIS IS THE MID-POINT IN THE 60 DEG SEGMENT
                     *
E2BA 96 C6           TMDPT1  LDA A  NPPMD          ;GET NO. FOR MID-POINT P.P.
E2BC 91 C5                   CMP A  NPPCT          ;COMPARE TO PULSE PERIOD COUNTEI
E2BE 26 08                   BNE    NMDPT1         ;BRANCH IF NOT MID-POINT
                     *
                     * MID-POINT OF 60 DEG SEGMENT ENCOUNTERED
                     *   COMPLEMENT 'J' INDEX (COUNT DOWN - ) TO ACCOUNT FOR
                     *   FOR COMPLEMENTARY PULSE WIDTH BECOMING 1ST COMP
                     * SET FTPTR INDEX TO COUNTDOWN
                     *
E2C0 73 0024                 COM    J              ;COMPLEMENT J INDEX
E2C3 CE FFFC                 LDX    #-04           ;SET FTPTR INDEX
E2C6 DF C8                   STX    FTINCR         ;STORE NEW INCREMENT VALUE
                     *
                     * UPDATE POINTERS AND INDICES AND TEST FOR END OF 60 DEG SEC
                     *
E2C8 73 0024         NMDPT1  COM    J              ;J = NOT J
E2CB 96 C3                   LDA A  FTPTR          ;LOAD PRESENT POSITION MS
E2CD D6 C4                   LDA B  FTPTR+1        ;  "       "        "    LS
E2CF DB C9                   ADD B  FTINCR+1       ;MOVE IN TABLE MS
E2D1 99 C8                   ADC A  FTINCR         ; "    "    "   LS
E2D3 97 C3                   STA A  FTPTR          ;STORE IN NEW POSITION MS
E2D5 D7 C4                   STA B  FTPTR+1        ; IN TABLE             LS
                     *
                     * TEST FOR END OF PULSE PERIODS
                     *
E2D7 7A 00C5                 DEC    NPPCT          ;DECR TOTAL NO. OF PP
E2DA 26 03                   BNE    MPPTD1         ;SKIP IF MORE PP TO GO
E2DC 73 00C0                 COM    F60DG          ;SET 60 DEG FLAG
                     *
                     * EVERY PULSE PERIOD WILL OUT SET/PRESET OR NULL
                     *  OPERATION COMMAND TO PIA TO SET UP FLIP FLOPS
                     *
E2DF 96 44           MPPTD1  LDA A  PIACMD         ;GET PIA CMD'D STATE
E2E1 B7 5002                 STA A  PIA+2          ;START CONV TO CLEAR DETECT
E2E4 86 72                   LDA A  #$72           ;PR/RST=1,NEW MUX ADDR-SPD CMD
E2E6 97 44                   STA A  PIACMD         ;CLEAR PIA CMD
                     *
                     * DISMISS THE INTERRUPT
                     *
E2E8 3B                      RTI
          E342               ORG    $E342
                     *
                     * COUNTERCLOCKWISE ROUTINE
                     *
E342 96 C5           CCWRT   LDA A  NPPCT          ;GET NO. OF PP COUNTED
E344 84 01                   AND A  #1             ;IS IT ODD?
E346 26 3A                   BNE    EVPP2          ;BRANCH FOR EVEN PP
                     *
                     * DOMINANT POLARITY DOES NOT EQUAL POLE VOLTAGE STATE
                     * DETERMINE THE DOMINANT PHASE (INIT: 0=A,1=B,2=C )
                     *
```

```
SEQ  LOC  OBJ      SOURCE

0553 E348 96 20              LDA A  DOMID           ;LOAD DOMINANT PHASE ID
0554 E34A 4A                 DEC A                  ;DECREMENT PHASE (B-A-C ALWAYS)
0555 E34B 27 24              BEQ    PHCD02          ;BRANCH B PHASE DOMINANT (ODD.
0556 E34D 2A 11              BPL    PHBD02          ;BRANCH C PHASE DOMINANT (ODD
0557                   *
0558                   * PHASE A IS DOMINANT
0559                   * DOMINANT POLARITY DOES NOT EQUAL POLE VOLTAGE STATE A-C-B
0560                   *
0561 E34F DE 78              LDX    T1              ;LOAD T1 COUNT
0562 E351 FF 4002            STX    PTM1+2          ;T1 TO PHASE A (DOM/1ST ON)
0563 E354 DE 7A              LDX    T2              ;LOAD T2 COUNT
0564 E356 FF 4006            STX    PTM1+6          ;T2 TO PHASE C(2ND ON/1ST OFF.
0565 E359 DE 7C              LDX    T3              ;LOAD T3 COUNT
0566 E35B FF 4004            STX    PTM1+4          ;T3 TO PHASE B (DOM/2ND OFF)
0567 E35E 20 5A              BRA    TMDPT2          ;BRANCH TO TEST FOR MID-POINT
0568                   *
0569                   * PHASE B IS DOMINANT  B-A-C
0570                   *
0571 E360 DE 78   PHBD02     LDX    T1              ;LOAD T1 COUNT
0572 E362 FF 4004            STX    PTM1+4          ;T1 TO PHASE B (DOM/1ST ON)
0573 E365 DE 7A              LDX    T2              ;LOAD T2 COUNT
0574 E367 FF 4002            STX    PTM1+2          ;T2 TO PHASE A(2ND ON/1ST OFF)
0575 E36A DE 7C              LDX    T3              ;LOAD T3 COUNT
0576 E36C FF 4006            STX    PTM1+6          ;T3 TO PHASE C (DOM/2ND OFF)
0577 E36F 20 49              BRA    TMDPT2          ;BRANCH TO TEST FOR MID-POINT
0578                   *
0579                   * PHASE C IS DOMINANT    C-B-A
0580                   *
0581 E371 DE 78   PHCD02     LDX    T1              ;LOAD T1 COUNT
0582 E373 FF 4006            STX    PTM1+6          ;T1 TO PHASE C (DOM/1ST ON)
0583 E376 DE 7A              LDX    T2              ;LOAD T2 COUNT
0584 E378 FF 4004            STX    PTM1+4          ;T2 TO PHASE B(2ND ON/1ST OFF)
0585 E37B DE 7C              LDX    T3              ;LOAD T3 COUNT
0586 E37D FF 4002            STX    PTM1+2          ;T3 TO PHASE A (DOM/2ND OFF)
0587 E380 20 38              BRA    TMDPT2          ;BRANCH TO TEST FOR MID-POINT
```

```
SEQ  LOC  OBJ       SOURCE

0589                *
0590                * DOMINANT POLARITY EQUALS POLE VOLTAGE STATE
0591                * DETERMINE THE DOMINANT PHASE
0592                *
0593 E382 96 20     EVPP2   LDA A  DOMID            ,LOAD DOMINANT ID
0594 E384 4A                DEC A                 ,DECREMENT PHASE ( -=A,0=B,+=C)
0595 E385 27 24             BEQ    PHCDE2             ;PHASE B DOMINANT - EVEN PP
0596 E387 2A 11             BPL    PHBDE2             ;PHASE C DOMINANT - EVEN PP
0597                *
0598                * PHASE A IS DOMINANT   B-C-A
0599 E389 DE 78             LDX    T1                 ;LOAD T1 COUNT
0600 E38B FF 4004           STX    PTM1+4             ;T1 TO PHASE B (DOM/2ND ON)
0601 E38E DE 7A             LDX    T2                 ;LOAD T2 COUNT
0602 E390 FF 4006           STX    PTM1+6             ;T2 TO PHASE C
0603 E393 DE 7C             LDX    T3                 ;LOAD T3 COUNT
0604 E395 FF 4002           STX    PTM1+2             ;T3 TO PHASE A
0605 E398 20 20             BRA    TMDPT2             ;BRANCH TO TEST FOR MID-POI
0606                *
0607                * PHASE B IS DOMINANT C-A-B
0608                *
0609 E39A DE 78     PHBDE2  LDX    T1                 ;LOAD T1 COUNT
0610 E39C FF 4006           STX    PTM1+6             ;T1 TO PHASE C
0611 E39F DE 7A             LDX    T2                 ;LOAD T2 COUNT
0612 E3A1 FF 4002           STX    PTM1+2             ;T2 TO PHASE A
0613 E3A4 DE 7C             LDX    T3                 ;LOAD T3 COUNT
0614 E3A6 FF 4004           STX    PTM1+4             ;T3 TO PHASE B
0615 E3A9 20 0F             BRA    TMDPT2             ;BRANCH TOTEST FOR MID-POIN
0616                *
0617                * PHASE C IS DOMINANT A-B-C
0618                *
0619 E3AB DE 78     PHCDE2  LDX    T1                 ;LOAD T1 COUNT
0620 E3AD FF 4002           STX    PTM1+2             ;T1 TO PHASE A
0621 E3B0 DE 7A             LDX    T2                 ;LOAD T2 COUNT
0622 E3B2 FF 4004           STX    PTM1+4             ;T2 TO PHASE B
0623 E3B5 DE 7C             LDX    T3                 ;LOAD T3 COUNT
0624 E3B7 FF 4006           STX    PTM1+6             ;T3 TO PHASE C
```

```
SEQ  LOC  OBJ       SOURCE

                    *
                    * TEST IF THIS IS THE MID-POINT IN THE 60 DEG SEGMENT
                    *
     E3BA 96 C6     TMDPT2  LDA A  NPPMD          ;GET NO. FOR MID-POINT P.P.
     E3BC 91 C5             CMP A  NPPCT          ;COMPARE TO PULSE PERIOD COUNTEL
     E3BE 26 08             BNE    NMDPT2          ;BRANCH IF NOT MID-POINT
                    *
                    * MID-POINT OF 60 DEG SEGMENT ENCOUNTERED
                    *   COMPLEMENT 'J' INDEX (COUNT DOWN - ) TO ACCOUNT FOR
                    *   FOR COMPLEMENTARY PULSE WIDTH BECOMING 1ST COMP
                    * SET FTPTR INDEX TO COUNTDOWN
                    *
     E3C0 73 0024           COM    J               ;COMPLEMENT J INDEX
     E3C3 CE FFFC           LDX    #-04            ;SET FTPTR INDEX
     E3C6 DF C8             STX    FTINCR          ;STORE NEW INCREMENT VALUE
                    *
                    * UPDATE POINTERS AND INDICES AND TEST FOR END OF 60 DEG SEG
                    *
     E3C8 73 0024 NMDPT2    COM    J               ;J = NOT J
     E3CB 96 C3             LDA A  FTPTR          ;LOAD PRESENT POSITION MS
     E3CD D6 C4             LDA B  FTPTR+1        ;  "       "        "    LS
     E3CF DB C9             ADD B  FTINCR+1       ;MOVE IN TABLE MS
     E3D1 99 C8             ADC A  FTINCR         ; "    "   "    LS
     E3D3 97 C3             STA A  FTPTR          ;STORE IN NEW POSITION MS
     E3D5 D7 C4             STA B  FTPTR+1        ; IN TABLE             LS
                    *
                    * TEST FOR END OF PULSE PERIODS
                    *
     E3D7 7A 00C5           DEC    NPPCT           ;DECR TOTAL NO. OF PP
     E3DA 26 03             BNE    MPPTD2          ;SKIP IF MORE PP TO GO
     E3DC 73 00C0           COM    F60DG           ;SET 60 DEG FLAG
                    *
                    * EVERY PULSE PERIOD WILL OUT SET/PRESET OR NULL
                    *  OPERATION COMMAND TO PIA TO SET UP FLIP FLOPS
                    *
     E3DF 96 44     MPPTD2  LDA A  PIACMD         ;GET PIA CMD'D STATE
     E3E1 B7 5002           STA A  PIA+2          ;START CONV TO CLEAR DETECT
     E3E4 86 72             LDA A  #$72           ;PR/RST=1,NEW MUX ADDR-SPD CMD
     E3E6 97 44             STA A  PIACMD         ;CLEAR PIA CMD
                    *
                    * DISMISS THE INTERRUPT
                    *
     E3E8 3B                RTI
                            END

0669 LINES ASSEMBLED,  LOC = E3E9,  0000 ERRORS DETECTED.
```

SYMBOL	VALUE	ATTR	LOCN	LINK
F60DG	00C0	84	2AA2	
FTPNW	00C1	84	2AAB	
FTPTR	00C3	84	2AB4	
NPPMD	00C6	84	2ABD	
BSTAR	00D2	84	2AC6	
SF	00D0	84	2ACF	
PWDOM	0072	84	2AD8	
PWCOM	0074	84	2AE1	
TPP	0070	84	2AEA	
T1	0078	84	2AF3	
T2	007A	84	2AFC	
T3	007C	84	2B05	
VPADD	00D8	84	2B0E	
SFMAX	0030	84	2B17	
SFMIN	0032	84	2B20	
PIACMD	0044	84	2B29	
DOMID	0020	84	2B32	
DPF	0022	84	2B3B	
J	0024	84	2B44	
FTINCR	00C8	84	2B4D	
MONFG	0051	84	2B56	
VSRAW	003A	84	2B5F	
VSFLG	00E7	84	2B68	
BPRAW	004A	84	2B71	
FCRAW	0038	84	2B7A	
ESTAR	003C	84	2B83	
NPPCT	00C5	84	2B8C	
FOUT	004C	84	2B95	
FCMD	0046	84	2B9E	
DIRCMD	0009	84	2BA7	
DIROUT	0010	84	2BB0	
MPIER	2000	84	2BB9	
MCAND	2002	84	2BC2	
PROD	2005	84	2BCB	
PIA	5000	84	2BD4	
PTM1	4000	84	2BDD	
PTM2	3000	84	2BE6	
WLATCH	2007	84	2BEF	
PTIS	E000	84	2BF8	
NOT60	E055	84	2C01	
AA	E036	84	2C0A	
A2	E041	84	2C13	
BPIN	E07D	84	2C1C	
CBPC	E084	84	2C25	
SVPADD	E09A	84	2C2E	
SFCAL	E0A0	84	2C37	
SFMNL	E0CC	84	2C40	
SFMSZ	E0C8	84	2C49	
SFOK	E0D0	84	2C52	
MINDX	E120	84	2C5B	
OTCNTS	E142	84	2C64	
DAN1	E156	84	2C6D	
DAN2	E15E	84	2C76	
DAN3	E161	84	2C7F	

SYMBOL	VALUE	ATTR	LOCN	LINK
CWRT	E242	84	2C88	
CCWRT	E342	84	2C91	
EVPP1	E282	84	2C9A	
PHBDO1	E260	84	2CA3	
PHCDO1	E271	84	2CAC	
TMDPT1	E2BA	84	2CB5	
PHBDE1	E29A	84	2CBE	
PHCDE1	E2AB	84	2CC7	
NMDPT1	E2C8	84	2CD0	
MPPTD1	E2DF	84	2CD9	
EVPP2	E382	84	2CE2	
PHCDO2	E371	84	2CEB	
PHBDO2	E360	84	2CF4	
TMDPT2	E3BA	84	2CFD	
PHCDE2	E3AB	84	2D06	
PHBDE2	E39A	84	2D0F	
NMDPT2	E3C8	84	2D18	
MPPTD2	E3DF	C4	2D21	

END ASM V1.1

```
AMI 6800
ASM V1.1

SEQ  LOC  OBJ     SOURCE

0001                       NAM    FEEDBACK CONTROL MPU PROGRAM
0002                 * MP917
0003                 * NSRDC
0004           FFF8        ORG    $FFF8
0005           0001        LSYMB  1
0006 FFF8 FC 00     IRQV   FDB    $FC00        ;IRQ VECTOR, E*OUTPUT
0007 FFFA F8 00     SWIV   FDB    $F800        ;SWI VECTOR
0008 FFFC F8 00     NMIV   FDB    $F800        ;NMI VECTOR
0009 FFFE F8 00     RSTV   FDB    $F800        ;RESET VECTOR
0010           FCD0        ORG    $FCD0
0011                **CONSTANTS FOR FLTCL
0012                *
0013                * FILTER OF FORM - (S+W1)/(S+W2)
0014                *   - WHERE W1 = "ZERO" AT 180 HZ.
0015                *   -       W2 = "POLE" AT 0.18 HZ.
0016                *
0017                *  VALUES CALCULATED FOR DELTA T = 6.0 MS.
0018 FCD0 3F EE     AADD   FDB    16366        ;A CONSTANT (0.998920583)
0019 FCD2 DD 7D     BADD   FDB    -8835        ;B CONSTANT (-0.539275)
0020 FCD4 22 80     CADD   FDB    08832        ;C CONSTANT (0.53906317)
0021                * DEFINITIONS REQUIRED FOR SUBROUTINES& PROGRAM
0022                *
0023           0001 FREQ   EQU    $0001        ;FREQUENCY ADDRESS
0024           0002 IQAD   EQU    $0002        ;IQ ADDRESS
0025           0003 VDAD   EQU    $0003        ;VD ADDRESS
0026           0004 VQAD   EQU    $0004        ;VQ ADDRESS
0027           0005 IDAD   EQU    $0005        ;ID ADDRESS
0028           0012 ERMT   EQU    $0012        ;ERROR MS BYTE
0029           0013 ERLT   EQU    $0013        ;ERROR LS BYTE
0030           C000 LLS    EQU    $0C000       ;LATCH ERROR LS BYTE
0031           B000 LMS    EQU    $0B000       ;LATCH ERROR MS BYTE
0032           0012 MPRM   EQU    $12          ;MULTIPLIER MS BYTE
0033           0013 MPRL   EQU    $13          ;MULTIPLIER LS BYTE
0034           0010 MPDM   EQU    $10          ;MULTIPLICAND MS BYTE
0035           0011 MPDL   EQU    $11          ;MULTIPLICAND LS BYTE
0036           0011 ADDL   EQU    $11          ;ADDEND LS BYTE
0037           0010 ADDM   EQU    $10          ;ADDEND MS BYTE
0038           0013 SUML   EQU    $13          ;SUM LS BYTE
0039           0012 SUMM   EQU    $12          ;SUM MS BYTE
0040           00FD RADD   EQU    $0FD         ;R* CORRECTION K-MS ADDR
0041           00FE KADD   EQU    $0FE         ;2*PI*F*L1 ADDR MS BYTE
0042           0010 PROD   EQU    MPDM         ;MS BYTE-PRODUCT
0043                *PRODUCT IS 4 BYTES FROM PROD TO PROD+3
0044           0010 DIFF   EQU    MPDM         ;DIFFERENCE MS BYTE
0045                * DIFFERENCE IS 2 OR 4 BYTES FROM DIFF TO DIFF+3
0046           0032 ESTAR  EQU    $32          ;E* AFTER FLXCL @32,33
0047           0021 AXMS   EQU    $21          ;A*XN BYTES(21-25)
0048           0026 BUN1M  EQU    $26          ;B*UN1 MS BYTES
0049           0028 BUN1LM EQU    $28          ;B*UN1 LS MID & LS
0050           002A CUNM   EQU    $2A          ;C*UN MS BYTES
0051           002C CUNLM  EQU    $2C          ;C*UN LS MID & LS
0052           0032 UN1    EQU    $32          ;UN+1 2  BYTES
0053           0047 XN     EQU    $47          ;XN - 3  BYTES
0054           0034 UN     EQU    $34          ;UN - 2  BYTES
0055           002A CMB    EQU    CUNM         ;C*UN-B*XN1
```

```
SEQ   LOC   OBJ       SOURCE

0056              0026 XN1T    EQU     BUN1M          ;XN1 TEMPORARY LOC
0057              0040 OLDF    EQU     $40            ;OLD FREQ VALUE
0058              0041 NEWF    EQU     $41            ;NEW FREQ VALUE
0059              0043 TMCNT   EQU     $43            ;TIME INTERVAL TWEEN ERRST'S
```

```
SEQ  LOC  OBJ    SOURCE

0061             **** SCALING COMMENTS FOR PROGRAM ****
0062             *     EXISTING LOAD TRIAL
0063             *WORD SCALING INTERNAL TO PROGRAM
0064             *
0065             *********FORMATS******
0066             *
0067             * B  B  B  B  B  B  B  B  B  B  B  B  B  B  B  B
0068             *31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16
0069             * B  B  B  B  B  B  B  B  B  B  B  B  B  B  B  B
0070             *15 14 13 12 11 10 09 08 07 06 05 04 03 02 01 00
0071             *
0072             *VD & VQ
0073             *S  64 32 16  8  4  2  1.0   0  0  0  0  0  0  0
0074             *ID &IQ
0075             *S 128 64 32  16 8  4  2 0.  0  0  0  0  0  0  0
0076             *
```

```
SEQ  LOC  OBJ      SOURCE

0078                *
0079          F800          ORG    $F800
0080                * INITIALIZATION SEQUENCE (ALL VECTORS POINT HERE)
0081                *  SEQUENCE ACCOMPLISHES THE FOLLOWING :
0082                *
0083                *    STACK POINTER SET TO TOP OF RAM (ADDR = 03FF)
0084                *    PIA IS SET UP
0085                *      PA = INPUTS (A/D'S OR FREQ)
0086                *      PB = OUTPUTS
0087                *        PB INITIAL VALUES PB7-PB4 = 0; NO A/D INPUTS ENABLED
0088                *                          PB1     = 1 NO FREQ INPUT ENABLED
0089                *                          PB0     = 0 NO START CONV COMMAND
0090                *    ERROR LATCH SET AT ALL "0'S"
0091                *
0092                *
0093                *
0094 F800 8E 03FF ISTART  LDS    #$03FF       ;SET TOP OF STACK
0095 F803 86 00   PAS     LDA A  #$00         ;SET CRA WORD; DDRA CHOSEN
0096 F805 CE 0000         LDX    #$0000       ;START ADDR
0097 F808 A7 00   PAS1    STA A  0,X          ;CLEAR RAM BYTE
0098 F80A 08              INX                 ;INCR BYTE COUNT
0099 F80B 8C 0100         CPX    #$0100       ;TEST FOR LAST ADDR
0100 F80E 26 F8           BNE    PAS1         ;BRANCH FOR MORE ADDR
0101 F810 B7 D001         STA A  $D001        ;STORE IN CRA, DDRA ENABLED
0102 F813 B7 D000         STA A  $D000        ;MAKE PA ALL INPUTS
0103 F816 86 04           LDA A  #$04         ;SET CRA WD, PRA CHOSEN
0104 F818 B7 D001         STA A  $D001        ;ENABLE PRA FOR READ
0105 F81B 86 00   PBS     LDA A  #$00         ;SET CRB WD, DDRB CHOSEN
0106 F81D B7 D003         STA A  $D003        ;SET CRB, DDRB ENABLED
0107 F820 86 FF           LDA A  #$FF         ;SET DDRB WD
0108 F822 B7 D002         STA A  $D002        ;SET DDRB TO ALL OUTPUTS
0109 F825 86 04           LDA A  #$04         ;SET CRB WD, PRB CHOSEN
0110 F827 B7 D003         STA A  $D003        ;STORE IN CRB, PRB ENABLED
0111 F82A 86 02           LDA A  #$02         ;LD PRB WD
0112 F82C B7 D002         STA A  $D002        ;SET PRB (NOTHING GOING)
0113 F82F 86 00   EVS     LDA A  #$00         ;SET INIT ERROR VALUE
0114 F831 B7 B000         STA A  LMS          ;STORE ERROR WD MS 6
0115 F834 B7 C000         STA A  LLS          ;STORE ERROR WD LS 6
0116 F837 86 FE           LDA A  #KADD        ;LD CONSTANT ADDR MSB
0117 F839 97 00           STA A  $00          ;STORE ADDR
0118 F83B 86 01           LDA A  #$01           ;LD TIME INTERVAL LS
0119 F83D 97 44           STA A  TMCNT+1        ;STORE LS OF INTERVAL
0120 F83F 0E              CLI                 ;CLEAR INT MASK
```

```
LOC  OBJ      SOURCE

                * MAIN SEQUENCE PROGRAM
                *  INPUT OF DATA (F,VD,ID,VQ,IQ) IS FIRST
                *  FLUX CALCULATION IS NEXT
                *  FILTER CALCULATION IS THIRD
                *  OUTPUT OF DATA WORD IS FINAL OPERATION
                *   LOOP BACK TO INPUT NEXT SAMPLE
                *
F840 BD F85F INPUT   JSR    INP          ;JSR INPUT ROUTINE
F843 BD FB41         JSR    RECTFY       ;JSR RECTIFY ID FOR OVERLOAD PRO
F846 BD F9CA         JSR    FLTCL        ;JSR FILTER CALCULATION
F849 BD FAD7         JSR    ERRST        ;JSR OUTPUT ROUTINE
F84C 7E F840         JMP    INPUT        ;JUMP & CONTINUE
                *
                *        GLOBALS ARE :
                *        INP - INPUT ROUTINE
                *        FLXCL - FLUX ROUTINE
                *        FLTCL - FILTER ROUTINE
                *        ERRST - ERROR WORD OUTPUT ROUTINE
```

```
LOC  OBJ       SOURCE

               *
               ************ SUBROUTINES ***************
               *
               * INP    (INPUT OF FREQ, VOLTAGE, & CURRENT)
               *         (F, VD, VQ, ID, IQ)
               *  READS FREQ IN PA
               *  DOES A/D CONVERSION
               *  STORES ALL VALUES DIRECT
               *   MEMORY WHERE VALUES STORED IS :
               *     ADDR           CONTENTS
               *    0000              "FE"
               *    0001               F
               *    0002              IQ
               *    0003              VD
               *    0004              VQ
               *    0005              ID
               *
               *  CONTROL OF PIA IS VIA PORT B
               *   THESE WORDS ARE RECOGNIZED *
               *  B7 B6 B5 B4 B3 B2 B1 B0
               *   0  0  0  0  X  X  1  0   RESET VALUE
               *   0  0  0  0  X  X  1  1   START CONVERSION
               *   0  0  0  1  X  X  1  0   READ IQ
               *   0  0  1  0  X  X  1  0   READ VD
               *   0  1  0  0  X  X  1  0   READ VQ
               *   1  0  0  0  X  X  1  0   READ ID
               *   0  0  0  0  X  X  0  0   READ FREQUENCY
               *
               *F AT INPUT IS B6 B5 B4 B3 B2 B1 X  X
               ***
          F85F          ORG    $F85F
          0055 FZFG     EQU    $55           ;FREQ =0 FLAG
F85F 86 00     INP      LDA A  #$00          ;LD READ FREQ WD
F861 B7 D002            STA A  $D002         ;STORE IN PRB
F864 B6 D000            LDA A  $D000         ;READ FREQ IN PRA
F867 43                 COM A                ;F WAS INVERTED
F868 44                 LSR A                ;SCALE FREQUENCY
F869 44                 LSR A                ;SCALE FREQUENCY
               *TEST FOR FREQ=0, IF YES HOLD E*=0 TO FILTER
F86A 81 00              CMP A  #$00          ;TEST FOR F=0
F86C 26 06              BNE    CNFI          ;BRANCH IF NOT 0
F86E C6 01              LDA B  #$01          ;ESTABLISH FLAG VALUE
F870 D7 55              STA B  FZFG          ;SET FLAG
F872 20 03              BRA    CNFI1         ;BRANCH TO CONT INP
F874 7F 0055   CNFI     CLR    FZFG          ;RESET F=0 FLAG
F877 97 41     CNFI1    STA A  NEWF          ;STORE NEW FREQ
F879 80 06              SUB A  #06           ;GEN F OFFSET
F87B 2A 02              BPL    GT6           ;BRANCH IF >6
F87D 86 00              LDA A  #$00          ;LD MIN OFFSET=6HZ
F87F 48        GT6      ASL A                ;POSITION OFFSET
F880 8B 06              ADD A  #06           ;FREQ POINTER
F882 97 01              STA A  FREQ          ;STORE VALUE
F884 86 03              LDA A  #$03          ;SET START CONV WD
F886 C6 02              LDA B  #$02          ;SET SC TURN OFF WD
```

```
SEQ  LOC  OBJ       SOURCE

0195 F888 B7 D002          STA A  $D002      ;SET SC IN PRB
0196 F88B F7 D002          STA B  $D002      ;TURN OFF SC
0197 F88E 86 12            LDA A  ##$12      ;LD READ IQ WD
0198 F890 C6 18            LDA B  ##$18      ;LD LOOP CNT(MUST > 80MS)
0199 F892 5A        ADLOOP DEC B             ;DECR LOOP CNT(WAIT ON A/D)
0200 F893 26 FD            BNE    ADLOOP     ;JUMP IF A/D NOT DONE
0201 F895 B7 D002          STA A  $D002      ;ENABLE IQ AT PRA
0202 F898 F6 D000          LDA B  $D000      ;LD IQ
0203 F89B C8 7F            EOR B  ##$7F      ;2'S COMPL IQ
0204 F89D D7 02            STA B  IQAD       ;STORE IQ
0205 F89F 86 22            LDA A  ##$22      ;LD READ VD WORD
0206 F8A1 B7 D002          STA A  $D002      ;ENAB VD AT PRA
0207 F8A4 F6 D000          LDA B  $D000      ;LD VD
0208 F8A7 C8 7F            EOR B  ##$7F      ;2'S COMPL VD
0209 F8A9 D7 03            STA B  VDAD       ;STORE VD
0210 F8AB 86 42            LDA A  ##$42      ;LD READ VQ WORD
0211 F8AD B7 D002          STA A  $D002      ;ENAB VQ AT PRA
0212 F8B0 F6 D000          LDA B  $D000      ;LD VQ
0213 F8B3 C8 7F            EOR B  ##$7F      ;2'S COMPL VQ
0214 F8B5 D7 04            STA B  VQAD       ;STORE VQ
0215 F8B7 86 82            LDA A  ##$82      ;LD READ ID WORD
0216 F8B9 B7 D002          STA A  $D002      ;ENAB ID AT PRA
0217 F8BC F6 D000          LDA B  $D000      ;LD ID
0218 F8BF C8 7F            EOR B  ##$7F      ;2'S COMPL ID
0219 F8C1 D7 05            STA B  IDAD       ;STORE ID
0220 F8C3 86 02            LDA A  ##$02      ;LD RESET VALUE TO A/D
0221 F8C5 B7 D002          STA A  $D002      ;RESET A/D ENABLES
0222 F8C8 86 FE            LDA A  ##$FE      ;LD K1 MS ADDR
0223 F8CA 97 00            STA A  $0000      ;STORE K1 MS ADDR
0224 F8CC 39               RTS
```

```
LOC  OBJ       SOURCE

               *
               ************ FLXCL ************
               *  TRIES TO USE COMMON SECTION OF RAM
               *  FOR CALCULATION
               *  SEQUENCE OF OPERATIONS IS :
               *   LOOK UP 2*PI*L*F
               *   K1*IQ = P1
               *   VD-P1 = S1
               *   S1*IQ = P2  STORED AT 0020,0021
               *   K1*ID = P3
               *   VQ+P3 = S2
               *   S2*ID = P4
               *   P2-P4 = 2WX
               *   WXM=(VD-K1*IQ)IQ-(VQ+K1*ID)ID
               *   R-WX = ERR STORED AT 0010,0011
               *
               * THIS SECTION DOES P1,S1,P2
          F8D0          ORG     $F8D0
F8D0 DE 00     FLXCL    LDX     $0000            ;LD K1 ADDRESS
F8D2 A6 00              LDA A   0,X           ;LD K1 MS
F8D4 E6 01              LDA B   1,X           ;LD K1 LS
               *
               *K1 FORMAT (RANGE F=6-60,0.03769-0.3769)
               *  L1 ASSUMED = 1.0 MILLIHENRIES
               *B15 B14 B13 B12 B11 B10 B9 B8 B7 B6 B5 B4 B3 B2 B1 B0
               * S   1 . .5 .25  X   X   X  X  X  X  X  X  X  X  X  X
               *
F8D6 36        AB       PSH A                 ;TEMP STORE K1 MS
F8D7 37                 PSH B                 ;TEMP STORE K1 LS
               *P1 CALCULATION
F8D8 97 10              STA A   MPDM             ;STORE K1 FOR MULT
F8DA D7 11              STA B   MPDL             ;STORE K1 LS FOR MULT
F8DC C6 00              LDA B   #$00            ;LD IQ LS BYTE
F8DE 96 02              LDA A   IQAD            ;LD IQ MS
F8E0 97 12     PIQ      STA A   MPRM             ;STORE IQ FOR MULT
F8E2 D7 13              STA  B MPRL              ;STORE IQ LS
F8E4 BD FB58 AC         JSR     MPY16            ;DO K1*IQ
               *
               * K1*IQ FORMAT
               *  B31 B30 B29 B28 B27 B26------B01 B00
               *   S   S   S   S   64  32 . . . . . .
               *
               *P1 DONE,SCALE USES PROD AND PROD+1 FOR P1 AFTER SCALING
F8E7 86 02              LDA A   #02              ;LD LOOP COUNT
F8E9 78 0012 SCP1       ASL     PROD+2          ;SHIFT LS
F8EC 79 0011            ROL     PROD+1          ;SHIFT MID BYTE
F8EF 79 0010            ROL     PROD            ;SHIFT MS
F8F2 4A                 DEC A                  ;DECR LOOP CNT
F8F3 26 F4              BNE     SCP1            ;BRANCH FOR MORE SCALING
F8F5 96 10              LDA A   PROD             ;LD P1 MS
F8F7 D6 11              LDA B   PROD+1           ,LD PI LS,TRUNCATES
F8F9 43                 COM A                  ;SET -K1*IQ MS
F8FA 53                 COM B                  ;SET -K1*IQ LS
               *
```

```
LOC  OBJ          SOURCE

                  * K1*IQ FORMAT-ADJUSTED
                  * S 128 64 32 16 8 4 2 1 . .5 .25 X X X X---->
                  *  RANGE (0.0--->96.X)
                  *
F8FB 97 10                STA A  ADDM          ;STORE P1 MS,FOR ADD
F8FD D7 11                STA B  ADDL          ;STORE P1,LS
F8FF C6 00                LDA B  #$00         ;LD VD LS BYTE
F901 96 03                LDA A  VDAD         ;LD VD MS
                  * SCALING OF VD,MOVE RIGHT 1 PLACE ,K*ID MATCH
F903 47           CVD     ASR A               ;MOVE MS 1 PLACE
F904 56                   ROR B               ;MOVE LS 1PLACE
F905 0D                   SEC                 ;SET CRY FOR SUBT
                  *
                  * VD FORMAT ADJUSTED
                  * S S 64 32 16 8 4 2 1 . .5 .25 X X X X
                  * RANGE (+/- 128)
                  *
                  *DO (VD-K1*IQ); TRUNCATES TO 16 BITS
F906 BD FB8D      AD      JSR    ADD16         ;DO VD+P1
                  *S1 DONE
                  * S1 FORMAT SAME AS VD ADJUSTED
F909 C6 00                LDA B  #$00         ;LD IQ LS
F90B 96 02                LDA A  IQAD         ;LD IQ MS FOR ADD
F90D 97 10        SIC     STA A  MPDM          ;STORE VD MS FOR MULT
F90F D7 11                STA B  MPDL          ;STORE VD LS
                  *DO (VD-K1*IQ)*IQ
F911 BD FB58              JSR    MPY16         ;DO IQ*S1
                  *
                  * (VD-K1*IQ)*IQ FORMAT - 32 BITS
                  *  S X X X X X X X 8 4 2 1 . .5 .25 X X X---->LSB
                  *
                  *P2 DONE, STORE P2 FOR LATER USE
F914 96 10                LDA A  PROD          ;LD P2 MS
F916 D6 11        AE      LDA B  PROD+1        ;LD P2 LS
F918 97 20                STA A  $20          ;STORE P2 TEMP
F91A D7 21                STA B  $21          ;STORE P2 MS MID TEMP
F91C 96 12                LDA A  PROD+2        ;LD P2 LS MID
F91E D6 13                LDA B  PROD+3        ;LD P2 LS
F920 97 22                STA A  $22          ;STORE P2 LS MID
F922 D7 23                STA B  $23          ;STORE P2 LS
                  * SECTION DOES P3,S2,& P4 CALCULATION
F924 33           AF      PUL B               ;PULL K1 LS
F925 32                   PUL A               ;PULL K1 MS
                  *P3 CALCULATION
F926 97 10                STA A  MPDM          ;STORE K1 FOR MULT
F928 D7 11                STA B  MPDL          ;STORE K1 LS FOR MULT
F92A C6 00                LDA B  #$00         ;LD ID LS
F92C 96 05                LDA A  IDAD         ;LD ID MS
F92E 97 12        PID     STA A  MPRM          ;STORE ID FOR MULT
F930 D7 13                STA B  MPRL          ;STORE ID LS
F932 BD FB58              JSR    MPY16         ;DO K1*ID
                  *
                  * K1*ID FORMAT
                  *  B31 B30 B29 B28 B27 B26------B01 B00
                  *   S   S   S   S  64  32 ..............
```

```
SEQ  LOC  OBJ        SOURCE

0335                 *
0336                 *P3 DONE,SCALE USES PROD+1 AND PROD+2 FOR P3
0337 F935 86 02             LDA A  #02              ;LD LOOP COUNT
0338 F937 78 0012 SCP3      ASL    PROD+2           ;SHIFT LS
0339 F93A 79 0011           ROL    PROD+1           ;SHIFT MID BYTE
0340 F93D 79 0010           ROL    PROD             ;SHIFT MS
0341 F940 4A                DEC A                   ;DECR LOOP CNT
0342 F941 26 F4             BNE    SCP3             ;BRANCH FOR MORE SCALING
0343 F943 96 10             LDA A  PROD             ;LD P3 MS
0344 F945 D6 11             LDA B  PROD+1           ;LD P3 LS,TRUNCATES
0345                 *
0346                 * K1*ID FORMAT-ADJUSTED
0347                 * S 128 64 32 16 8 4 2 1 . .5 .25 X X X X---->
0348                 * RANGE (0.0--->96.X)
0349                 *
0350 F947 97 10             STA A  ADDM             ;STORE P3 MS,FOR ADD
0351 F949 D7 11             STA B  ADDL             ;STORE P3,LS
0352 F94B C6 00             LDA B  #$00             ;SET VQ LS= 0
0353 F94D 96 04             LDA A  VQAD             ;LD VQ MS
0354                 * SCALING OF VQ,MOVE RIGHT 1 PLACE ,K*IQ MATCH
0355 F94F 47         CVQ    ASR A                   ;MOVE MS 1 PLACE
0356 F950 56                ROR B                   ;MOVE LS 1PLACE
0357 F951 0C                CLC                     ;CLEAR CRY FOR ADD
0358                 *
0359                 * VQ FORMAT ADJUSTED
0360                 * S S 64 32 16 8 4 2 1 . .5 .25 X X X X
0361                 * RANGE (+/- 128)
0362                 *
0363                 *DO (VQ+K1*ID), TRUNCATES TO 16 BITS
0364 F952 BD FB8D           JSR    ADD16            ;DO VQ+P3
0365                 *S2 DONE
0366 F955 C6 00             LDA B  #$00             ;LD ID LS
0367 F957 96 05             LDA A  IDAD             ;LD ID MS FOR ADD
0368 F959 97 10      SID    STA A  MPDM             ;STORE VQ MS FOR MULT
0369 F95B D7 11             STA B  MPDL             ;STORE VQ LS
0370                 *DO (VQ+K1*ID)*ID
0371 F95D BD FB58           JSR    MPY16            ;DO ID*S2
0372                 *
0373                 * (VQ+K1*ID)*ID FORMAT - 32 BITS
0374                 *  S X X X X X X X 8 4 2 1 . .5 .25 X X X---->LSB
0375                 *
0376                 * SECTION DOES P2-P4 (32 BITS)
0377 F960 96 23      AK     LDA A  $23              ;LD P2 LS
0378 F962 D6 22             LDA B  $22              ;LD P2 LS MID
0379 F964 90 13             SUB A  DIFF+3           ;P2LS-P4LS
0380 F966 D2 12             SBC B  DIFF+2
0381 F968 97 13      AL     STA A  DIFF+3           ;STORE P2LS-P4LS
0382 F96A D7 12             STA B  DIFF+2           ;STORE P2LS,MID-P4LS,MID
0383 F96C 96 21             LDA A  $21              ;LD P2 MS MID
0384 F96E D6 20             LDA B  $20              ;LD P2 MS
0385 F970 92 11             SBC A  DIFF+1           ;P2MS,MID-P4MS,MID
0386 F972 D2 10             SBC B  DIFF             ;P2MS-P4MS
0387 F974 97 11      AM     STA A  DIFF+1           ;STORE P2MS,MID-P4MS,MID
0388 F976 D7 10             STA B  DIFF             ;STORE P2MS-P4MS
0389                 * WX IS 32 BITS NOW AT 0010 THRU 0013
```

```
SEQ  LOC  OBJ       SOURCE

0390                *
0391                * SECTION ADJUSTS 2WX FOR 16 BITS OF ACCURACY
0392                *  PRIOR TO "R" CORRECTION
0393                *     WX SIGNIFICANCE IS DIFF AND DIFF+1
0394                * R* & WX FORMAT IS 16 BITS
0395                * S 16K 8K 4K 2K 1K 512 256 128 64 32 16 8 4 2 1
0396                *
0397                *** WX IS POSITIVE ******************
0398                *
0399                *** TEST2 SCALING OF WX,MOVE LEFT 2 PLACES
0400 F978 96 55     WXCOR   LDA A  FZFG        ;LD F=0 FLAG
0401 F97A 27 0E             BEQ    SCWX                ;BRANCH IF F NOT 0
0402 F97C 4F                CLR A
0403 F97D 97 32             STA A  ESTAR             ;CLR ESTAR MS
0404 F97F 97 33             STA A  ESTAR+1           ;CLR ESTAR LS
0405 F981 97 34             STA A  UN
0406 F983 97 35             STA A  UN+1
0407 F985 97 47             STA A  XN                ;CLR XN MS
0408 F987 97 48             STA A  XN+1              ;CLR XN LS
0409 F989 39                RTS                ;RETURN FROM SUBR
0410 F98A C6 01     SCWX    LDA B  #$01        ;SET BIT COUNT FOR LOOP
0411 F98C 78 0013 SCWX1     ASL    DIFF+3          ;MOVE LS LEFT 1
0412 F98F 79 0012           ROL    DIFF+2          ;MOVE MID-LS LEFT 1
0413 F992 79 0011           ROL    DIFF+1          ;MOVE MID-MS LEFT 1
0414 F995 79 0010           ROL    DIFF            ;MOVE MS LEFT 1
0415 F998 5A                DEC B              ;DECR BIT COUNT
0416 F999 26 F1             BNE    SCWX1           ;BRANCH FOR MORE SCALING
0417                *
0418                * MODIFIED 3-30-83
0419                * (FILTERS WX BEFORE SUBTRACTING R*, CLAMPS WX)
0420                *
0421 F99B 96 10             LDA A  DIFF        ;GET WX MS
0422 F99D D6 11             LDA B  DIFF+1      ;GET WX LS
0423 F99F 43                COM A              ;COMPLEMENT
0424 F9A0 53                COM B
0425 F9A1 81 21             CMP A  #$21        ;CLAMP TO PREVENT FILTER O/F
0426 F9A3 2D 02             BLT    NOCLMP
0427 F9A5 86 20             LDA A  #$20
0428 F9A7 97 32     NOCLMP  STA A  ESTAR       ;STORE FOR FILTER
0429 F9A9 D7 33             STA B  ESTAR+1
0430 F9AB 39                RTS
0431                *
0432                * END OF MODIFICATION
0433                *
```

```
SEQ  LOC  OBJ       SOURCE

0435                *********** FLTCL **********
0436                * FILTER CALCULATION
0437                * PERFORMS S+A/S+B FILTER ALGORITHM
0438                * DONE FOR EVERY INPUT VALUE
0439                * INPUT IS 16 BITS STORED AT 0010,0011
0440                * OPERATIONS INCLUDE :
0441                *  XN1 = A*X - B*UN1 + C*U
0442                *  YN1 = UN1 + XN1
0443                *  CONSTANTS A,B,C ARE STORED IN PROM
0444                *  X,UN1 ARE STORED IN RAM AT 0030-0031,0032-0033
0445                *  UN1= E* FROM FLXCL
0446                *  U IS STORED AT 0034,0035
0447                *
0448                * FORMAT FOR DATA THIS SECTION
0449                *  CONSTANTS A,B,C HAVE 16 BITS AS
0450                * B15 B14 B13 B12 B11 B10 B9 B8 B7 B6 B5 B4 B3 B2 B1 B0
0451                *  S   1 . .5 .25  X   X   X  X  X  X  X  X  X  X  X  X
0452                *UN & UN+1 HAVE FORM
0453                *  S  16K  8K  4K  2K  1K 512 X  X 64 32 16  8  4  2  1
0454                *XN FORM
0455                *  S 4160K  X   X   X   X  X  X  X  X  X  X  X  X  X 256
0456                *XN+1 HAS FORM
0457                *  S 4160K  X   X   X   X  X  X  X  X  X  X  X  X  X 256
0458                *YN+1 HAS FORM
0459                *  S 4160K  X   X   X   X  X  X  X  X  X  X  X  X  X 256
0460                *
0461           F9CA           ORG    $F9CA
0462 F9CA FE FCD0 FLTCL      LDX    AADD          ; GET A
0463 F9CD DF 10              STX    MPDM          ; AND SAVE FOR MPY
0464 F9CF DE 47              LDX    XN            ; GET XN
0465                * SPECIAL OP FOR - XN'S
0466 F9D1 96 47              LDA A  XN            ;LD XN MS
0467 F9D3 49                 ROL A                ;GET SIGN
0468 F9D4 24 01              BCC    CXP           ;BRANCH FOR +XN
0469 F9D6 08                 INX                  ;INCR MS BYTE
0470 F9D7 DF 12      CXP     STX    MPRM          ; AND SAVE FOR MPY
0471 F9D9 BD FB58            JSR    MPY16            ;DO A*XN
0472                * A*XN DONE, STORE 4 BYTES OF PRODUCT
0473                *  STORE A*XN (40BITS) FOR ADD
0474 F9DC DE 10              LDX    PROD          ; GET A*XN MS 2
0475 F9DE DF 21              STX    AXMS          ;STORE MS 2
0476 F9E0 DE 12              LDX    PROD+2        ; STORE 2 LS BYTES
0477 F9E2 DF 23              STX    AXMS+2        ;STORE 2 LS
0478                * DOING 24 BY 16 BIT MULTIPLY
0479 F9E4 FE FCD0            LDX    AADD             ;LD A
0480 F9E7 DF 10              STX    MPDM             ;STORE FOR MULT
0481 F9E9 96 47              LDA A  XN            ;LD XN
0482 F9EB 49                 ROL A                ;GET SIGN
0483 F9EC 86 00      PXN     LDA A  #$00          ;LD + MS BYTE
0484 F9EE 24 02              BCC    CAXN             ;BRANCH FOR +XN
0485 F9F0 86 FF              LDA A  #$FF          ;LD - MS BYTE
0486 F9F2 D6 49      CAXN    LDA B  XN+2          ;LD XN LS BYTE
0487 F9F4 97 12              STA A  MPRM          ;STORE MS FOR MULT
0488 F9F6 D7 13              STA B  MPRM+1        ;STORE LS
```

```
SEQ  LOC  OBJ      SOURCE

0489 F9F8 BD FB58          JSR    MPY16        ;DO LS MULTIPLY
0490               * COMBINE PARTIAL PRODUCTS OF A*XN
0491 F9FB 96 13            LDA A  PROD+3       ;LD LS BYTE
0492 F9FD 97 25            STA A  AXMS+4       ;STORE LS BYTE
0493 F9FF 96 12            LDA A  PROD+2       ;LD LS MID
0494 FA01 D6 11            LDA B  PROD+1       ;LD MS MID
0495 FA03 9B 24            ADD A  AXMS+3       ;ADD LS MID
0496 FA05 D9 23            ADC B  AXMS+2       ;ADD MS MID
0497 FA07 97 24            STA A  AXMS+3       ;STORE LS MID
0498 FA09 D7 23            STA B  AXMS+2       ;STORE MS MID
0499 FA0B D6 10            LDA B  PROD         ;LD MS OF PROD
0500 FA0D 96 10            LDA A  PROD         ;LD MS OF PROD
0501 FA0F 99 22            ADC A  AXMS+1       ;ADD MS
0502 FA11 D9 21            ADC B  AXMS         ;ADD MS OF AXN
0503 FA13 97 22            STA A  AXMS+1       ;STORE MS MID
0504 FA15 D7 21            STA B  AXMS         ;STORE MS BYTE
0505               * HAVE DONE A*XN,NEXT OP IS B*UN1
0506 FA17 FE FCD2          LDX    BADD         ; GET B
0507 FA1A DF 10            STX    MPDM         ; SAVE FOR MPY
0508 FA1C DE 32            LDX    UN1          ; GET UN1
0509 FA1E DF 12            STX    MPRM         ; SAVE FOR MPY
0510 FA20 BD FB58          JSR    MPY16          ;DO B*UN1
0511 FA23 DE 10            LDX    PROD         ; SAVE B*UN1 IN LOCS 26,27,28,29
0512 FA25 DF 26            STX    BUN1M
0513 FA27 DE 12            LDX    PROD+2
0514 FA29 DF 28            STX    BUN1LM
0515               * HAVE COMPLETED B*UN1,NEXT OP IS C*U
0516 FA2B FE FCD4          LDX    CADD         ; GET C
0517 FA2E DF 10            STX    MPDM         ; SAVE FOR MPY
0518 FA30 DE 34            LDX    UN           ; GET UN
0519 FA32 DF 12            STX    MPRM         ; SAVE FOR MPY
0520 FA34 BD FB58          JSR    MPY16          ;DO C*U
0521 FA37 DE 10            LDX    PROD         ; SAVE C*UN IN LOCS 2A,2B,2C,2D
0522 FA39 DF 2A            STX    CUNM
0523 FA3B DE 12            LDX    PROD+2
0524 FA3D DF 2C            STX    CUNLM
0525               * HAVE COMPLETED MULT FOR XN1,NEXT OP IS XN1 CALC
0526               * DOING A*X-B*UN1+C*N.
0527               *
0528               * 1ST OPERATION - DO C*UN-B*UN1
0529               *  RESOLUTION NEEDED = 32 BITS
0530 FA3F 96 2D            LDA A  CUNLM+1      ;GET C*UN
0531 FA41 D6 2C            LDA B  CUNLM
0532 FA43 90 29            SUB A  BUN1LM+1     ;SUBTRACT: C*UN-B*UN1
0533 FA45 D2 28            SBC B  BUN1LM
0534               *  DON'T NEED LS BYTE DUE TO SCALING 8->
0535 FA47 97 2D            STA A  CMB+3        ;STORE LS(ALSO SCALES)
0536 FA49 D7 2C            STA B  CMB+2        ;STORE LS MID
0537 FA4B 96 2B            LDA A  CUNM+1       ;LD C*UN MS MID
0538 FA4D D6 2A            LDA B  CUNM         ;LD C*UN MS
0539 FA4F 92 27            SBC A  BUN1M+1      ;DO C*UN MS MID
0540 FA51 D2 26            SBC B  BUN1M        ;DO C*UN MS
0541 FA53 97 2B            STA A  CMB+1        ;STORE CUN-BUN1 LS MID
0542 FA55 D7 2A            STA B  CMB          ;STORE CUN-BUN1 MS MID
0543               * C*UN - B*UN1 DONE IN ADDR 2A->2D
```

```
LOC  OBJ       SOURCE

                    * DO A*AN+(C*UN-B*UN1)
                    *DOING SUM 40 BITS LS OF AXN =LS OF CMB
                    *
                    * FIRST MAKE CUN-BUN1 SAME LENGTH AS AX
                    *
          0050 CMBT     EQU     $50         ;SIGN OF CMB
FA57 96 2A              LDA A   CMB         ;LD CMB MS
FA59 49                 ROL A               ;GET SIGN
FA5A 86 00              LDA A   #$00        ;SET + SIGN
FA5C 24 02              BCC     SPCMB       ;BRANCH FOR +CMB
FA5E 86 FF              LDA A   #$FF        ;SET - SIGN
FA60 97 50     SPCMB    STA A   CMBT        ;STORE SIGN BYTE
                    *SIGN BYTE READY DO ADD
                    *
FA62 96 25              LDA A   AXMS+4      ;LD MS BYTE CUN-BUN1
FA64 9B 2D              ADD A   CMB+3       ;ADD LS BYTES
FA66 97 25              STA A   AXMS+4      ;TEMP STORE XN1 LS
FA68 96 24              LDA A   AXMS+3      ;LD A*XN LS
FA6A D6 23              LDA B   AXMS+2      ;LD A*XN LS MID
FA6C 99 2C              ADC A   CMB+2       ;ADD AX +CU-BU1 LS
FA6E D9 2B              ADC B   CMB+1
FA70 97 29              STA A   XN1T+3      ;STORE XN1 LS TEMP
FA72 D7 28              STA B   XN1T+2
FA74 96 22              LDA A   AXMS+1      ;LD A*X MS MID
FA76 D6 21              LDA B   AXMS        ;LS A*XN MS
FA78 99 2A              ADC A   CMB         ;ADD MS BYTES
FA7A D9 50              ADC B   CMBT        ;ADD SIGN CMB
FA7C 97 27     STXN1    STA A   XN1T+1      ;STORE XN1 MS MID
FA7E D7 26              STA B   XN1T        ;STORE MS TEMP
                    * XN1 DONE NEED TO SCALE TO MATCH UN1 & STORAGE
                    *  TO XN POSITION(SHIFT <- 2)
FA80 86 02              LDA A   #$02        ;LD SHIFT COUNT
FA82 78 0025   CXSCL    ASL     AXMS+4      ;SCALE XN1 LS
FA85 79 0029            ROL     XN1T+3      ;SCALE XN1
FA88 79 0028            ROL     XN1T+2      ;  IN
FA8B 79 0027            ROL     XN1T+1      ;  TEMP
FA8E 79 0026            ROL     XN1T        ;LOCAL
FA91 4A                 DEC A               ;DECR SHIFT COUNT
FA92 26 EE              BNE     CXSCL       ;BRANCH FOR MORE SCALE
                    *READY FOR YN+1
                    *  STORE UN & XN VALUE FOR NEXT CALC
FA94 DE 26              LDX     XN1T        ;LD XN1 MS + MID
FA96 96 28              LDA A   XN1T+2      ;LD LS MID=LS AFTER SCALE
FA98 DF 47              STX     XN          ;STORE XN1 AS XN NEXT CALC
FA9A 97 49              STA A   XN+2        ;STORE XN1 LS AS XN
FA9C DF 27              STX     XN1T+1      ;STORE XN1 SCALING
FA9E 97 29              STA A   XN1T+3      ;STORE XN1 LS SCALED
FAA0 96 33              LDA A   UN1+1       ;LD LS UN+1
FAA2 D6 32              LDA B   UN1         ;LD MS UN+1
FAA4 97 35              STA A   UN+1        ;STORE UN1 AS UN NEXT CALC
FAA6 D7 34              STA B   UN          ;STORE UN1 AS UN MS
                    *DOING YN+1=XN+1 + UN+1
FAA8 59                 ROL B               ;ROLL MS FOR SIGN
FAA9 D6 32              LDA B   UN1         ;RESTORE UN1
FAAB 24 0D              BCC     PUN1        ;BRANCH FOR + UN1
```

```
SEQ  LOC  OBJ      SOURCE

0599 FAAD 9B 29             ADD A  XN1T+3      ;ADD XN1+UN1 LS
0600 FAAF D9 28             ADC B  XN1T+2      ;ADD LS MID
0601                 *DON'T NEED LS OF YN+1
0602 FAB1 D7 13             STA B  ERLT        ;STORE LS OF E*
0603 FAB3 86 FF             LDA A  #$FF        ;SET NEG MS
0604 FAB5 99 27             ADC A  XN1T+1      ;ADD XN1+UN1 MS
0605 FAB7 97 12             STA A  ERMT        ;STORE E* MS
0606 FAB9 39                RTS
0607 FABA 9B 29      PUN1   ADD A  XN1T+3      ;ADD LS
0608 FABC D9 28             ADC B  XN1T+2      ;ADD XN1+UN1 LS MID
0609                 *DON'T NEED LS OF YN+1
0610 FABE D7 13             STA B  ERLT        ;STORE E* LS
0611 FAC0 86 00             LDA A  #$00        ;SET MS =0 FOR +
0612 FAC2 99 27             ADC A  XN1T+1      ;ADD MS BYTES
0613 FAC4 97 12             STA A  ERMT        ;STORE E* MS
0614 FAC6 39                RTS
```

```
SEQ  LOC  OBJ       SOURCE

                    *********** ERRSR ************
                    * ERROR WORD OUTPUT SUBROUTINE
                    *  OUTPUTS 12 BITS OF ERROR WORD TO MPA
                    *  TAKES 16 BITS & ADJUSTS FORMAT AS FOLLOWS:
                    *ERROR WORD
                    *B15 B14 B13 B12 B11 B10 B9 B8 B7 B6 B5 B4 B3 B2 B1 B0
                    * TWO BYTES OUTPUT TO LATCHES
                    * ADDR       B07 B06 B05 B04 B03 B02 B01 B00
                    * ERMT        X   X  B15 B14 B13 B12 B11 B10  STORE TO C000
                    * ERLT        X   X  B09 B08 B07 B06 B05 B04  STORE TO B000
                    *
                    ******
          003A ESOTM      EQU    $3A           ;E* MS VALUE FOR OUTPUT
          003B ESOTL      EQU    $3B           ;E* LS VALUE FOR OUTPUT
                    *
          FAD7            ORG    $FAD7
FAD7 96 41     ERRST      LDA A  NEWF          ;LD NEW FREQ CMD
FAD9 98 40                EOR A  OLDF          ;COMP TO OLD FREQ
FADB 27 0A                BEQ    NFCHG         ;BRANCH NO FREQ CHANGE
FADD CE 0001              LDX    #$0001        ;LD MAX TIME INTRVL
FAE0 96 41     SFTD       LDA A  NEWF          ;RELOAD NEW FREQ
FAE2 97 40                STA A  OLDF          ;STORE AS OLD FREQ
FAE4 DF 43     STCNT      STX    TMCNT         ;STORE MAX INTRVL
FAE6 39                   RTS                  ;RETURN
FAE7 DE 43     NFCHG      LDX    TMCNT         ;LOAD TIME CNT
FAE9 09                   DEX                  ;DECR TIME
FAEA 26 F8                BNE    STCNT         ;BRANCH IF MORE TIME
FAEC CE 0001              LDX    #$0001        ;SET TIME INTRVL TO MIN
FAEF DF 43                STX    TMCNT         ;STORE MIN TIME
FAF1 96 12                LDA A  ERMT          ;LOAD ERROR MS 8
FAF3 D6 13                LDA B  ERLT          ;LOAD ERROR LS 8
FAF5 7E FB30              JMP    BYPASS        ;BYPASS THE GP.. ROUTINE
                    *
                    *ALTERED CODE
                    *(PROVIDES FOR A VARIABLE GAIN FOR WX)
                    *
FAF8 86 FF                LDA A  #$FF          ;GET GAIN FROM TABLE
FAFA 97 00                STA A  00
FAFC DE 00                LDX    0000
FAFE A6 00                LDA A  0,X
FB00 E6 01                LDA B  1,X
FB02 97 10                STA A  $10           ;STORE FOR MULPIFLY
FB04 D7 11                STA B  $11
FB06 BD FB58              JSR    MPY16         ;GAIN X ESTAR
FB09 CE 0003              LDX    #$0003        ;SCALE DOWN PRODUCT
FB0C 77 0010   SCALE      ASR    $10
FB0F 76 0011              ROR    $11
FB12 76 0012              ROR    $12
FB15 76 0013              ROR    $13
FB18 09                   DEX
FB19 26 F1                BNE    SCALE
FB1B DE 12                LDX    $12           ;GET RESULT
FB1D DF 10                STX    $10           ;STORE FOR ADD
                    *
```

```
SEQ  LOC  OBJ         SOURCE

0670 FB1F 86 FD                LDA A   #RADD          ;LD R MS OF ADDR
0671 FB21 97 00                STA A   $00            ;CORRECT ADDR
0672 FB23 DE 00                LDX     $0000           ;LD R ADDRESS
0673 FB25 A6 00                LDA A   0,X            ;LD R MS
0674 FB27 E6 01                LDA B   1,X            ;LD R LS
0675 FB29 BD FB8D              JSR     ADD16           ;DO R-(+WX)
0676                  *
0677                  * END OF MODIFICATION 3-30-83
0678                  *
0679                  ** E* FOR FLTCL NOW EXISTS
0680                  * SCALING  S 16K 8K 4K 2K 1K ------ 16 8 4 2 1
0681 FB2C 96 12                LDA A   SUMM           ;LD ANS MS(R-WX)
0682 FB2E D6 13                LDA B   SUML           ;LD ANS LS(R-WX)
0683                  *FORMAT E*OUT FOR LATCH
0684 FB30 CE 0002 BYPASS  LDX     #$0002                ;LD SHIFT COUNT
0685 FB33 47      CES     ASR A                  ;SCALE MS
0686 FB34 56                   ROR B                  ;ROTATE LS RIGHT 1
0687 FB35 09                   DEX                    ;DECR LOOP CNT
0688 FB36 26 FB                BNE     CES                   ;BRANCH FOR MORE SCALING
0689                  * MS BYTE NOW CORRECT
0690                  *  FINISH FORMAT FOR ERLT TO LATCH
0691 FB38 57                   ASR B                  ;ROTATE LS RIGHT 1
0692 FB39 57                   ASR B                  ;ROTATE LS RIGHT 1
0693 FB3A 0F                   SEI                    ;SET MASK OF IRQ
0694 FB3B 97 3A                STA A   ESOTM            ;STORE MS TO RAM
0695 FB3D D7 3B                STA B   ESOTL            ;STORE LS TO RAM
0696 FB3F 0E                   CLI                    ;CLEAR INT MASK
0697 FB40 39                   RTS
0698                  *
0699                  * ID RECTIFIER ROUTINE FOR OVERLOAD PROTECTION  5-3-83
0700                  *
0701 FB41 D6 05   RECTFY  LDA B   IDAD           ;GET DIRECT CURRENT
0702 FB43 2A 01                BPL     POS            ;BRANCH IF POSITIVE
0703 FB45 50                   NEG B                  ;NEGATE IF NEGATIVE
0704 FB46 D7 33   POS     STA B   ESTAR+1        ;STORE IDAD
0705 FB48 7F 0032              CLR     ESTAR          ;CLEAR ESTAR MS BYTE
0706 FB4B CE 0004              LDX     #$0004         ;SCALE RECTIFIED ID FOR FILTER
0707 FB4E 78 0033 GAIN1   ASL     ESTAR+1
0708 FB51 79 0032              ROL     ESTAR
0709 FB54 09                   DEX
0710 FB55 26 F7                BNE     GAIN1
0711 FB57 39                   RTS
0712                  *
0713                  *END OF RECTIFIER ROUTINE  5-3-83
0714                  *
```

```
SEQ  LOC  OBJ         SOURCE

                      ********** MPY16 **************
                      * 16X16 BIT MULTIPLY SUBROUTINE
                      * SUBROUTINE IS JSRED AFTER FOLLOWING SETUP
                      *   MEMORY LOCATIONS
                      *     ADDR              CONTENTS
                      *     0010              MULTIPLICAND MS/PR MS
                      *     0011              MULTIPLICAND LS/PR MS-MID
                      *     0012              MULTIPLIER   MS/PR LS-MID
                      *     0013              MULTIPLIER   LS/PR LS
                      * INDEX REGISTER CONTAINS LOOP COUNT DURING SUBROUTINE
                      *
                      * TEMP STORAGE OF MULTIPLIER SIGN = $0014
               0014 MPIER   EQU    $0014
                      *
     FB56 CE 0010 MPY16   LDX    #0016          ;LOAD LOOP COUNT
     FB59 96 12           LDA A  MPRM           ;LD MULTIPLIER SIGN BYTE
     FB5D 97 14           STA A  MPIER          ;STORE SIGN BYTE
     FB5F 4F              CLR A                 ;CLEAR ACCUM
     FB60 5F              CLR B                 ;CLEAR ACCUM B
     FB61 76 0012         ROR    PROD+2         ;ROTATE MULTR MS8
     FB64 76 0013         ROR    PROD+3         ;ROTATE MULTR LS8
     FB67 24 06   SMPY1   BCC    SMPY2          ;BRANCH IF MULTR IS EVEN
     FB69 DB 11           ADD B  MPDL           ;ADD MULTD LS 8
     FB6B 99 10           ADC A  MPDM           ;ADD MULTD MS 8
     FB6D 29 02           BVS    SMPY3          ;BRANCH,ARITH OVRFL
     FB6F 47      SMPY2   ASR A                 ;EXTEND SIGN IF NOT
     FB70 49              ROL A                 ;RESTORE POSITION
     FB71 46      SMPY3   ROR A                 ;SHIFT PART PROD MS8(RIGHT)
     FB72 56              ROR B                 ;"     "    "   LS8   "
     FB73 76 0012         ROR    PROD+2         ;SHIFT PR&MULTR IN MEM
     FB76 76 0013         ROR    PROD+3         ;"   "   "     "  "
     FB79 09              DEX                   ;DECR LOOP  COUNT
     FB7A 26 EB           BNE    SMPY1          ;BRANCH IF NOT LAST BIT
     FB7C 77 0014         ASR    MPIER          ;TST SIGN OF MULTIPLIER
     FB7F 79 0014         ROL    MPIER          ;
     FB82 24 04           BCC    SMPY4          ;BRANCH SIGN +
     FB84 D0 11           SUB B  MPDL           ;CORRECT MID BYTE
     FB86 92 10           SBC A  MPDM           ;CORRECT MS BYTE
     FB88 D7 11   SMPY4   STA B  PROD+1         ;STORE PROD MS MID8
     FB8A 97 10           STA A  PROD           ;STORE PROD MS8
     FB8C 39              RTS
```

```
SEQ  LOC  OBJ      SOURCE

                   ********** ADD16 ****************
                   * 16 BIT ADD SUBROUTINE
                   *  ASSUMES ADDEND IS IN A,B REGISTERS
                   *   ADDER LOCATED IN 0010,0011 PRIOR TO JSR
                   *   SUM IS STORED IN 0012,0013
                   *   A = N MSB
                   *   B = N LSB
                   *  0010 = M MSB
                   *  0011 = M LSB
                   * AFTER ADD,
                   *  0012 = SUM MSB
                   *  0013 = SUM LSB
                   *
     FB8D D9 11    ADD16   ADC B  ADDL          ;ADD TWO LS BYTES
     FB8F 99 10            ADC A  ADDM          ;ADD TWO MS BYTES
     FB91 D7 13            STA B  SUML          ;STORE LS BYTE
     FB93 97 12            STA A  SUMM          ;STORE MS BYTE
     FB95 39               RTS
```

```
SEQ  LOC  OBJ       SOURCE

0777                ****** E* OUTPUT ROUTINE *****
0778                * WRITES TO E* LATCH IN RESPONSE TO INTERRUPT
0779                * INTERRUPT FROM MPA HAPPENS APPROX. 200MICROSEC
0780                *  PRIOR TO V* INPUT INTO MPA
0781                *
0782 FB96 96 3A     EOUT     LDA A   ESOTM         ;LD E* MS FOR LATCH
0783 FB98 D6 3B              LDA B   ESOTL         ;LD E* LS FOR LATCH
0784 FB9A B7 8000            STA A   LMS           ;STORE MS 8 TO LATCH
0785 FB9D F7 C000            STA B   LLS           ;STORE LS 8 TO LATCH
0786 FBA0 3B                 RTI                   ;RETURN FROM INT
0787                         END

0787 LINES ASSEMBLED,  LOC = FBA1,  0000 ERRORS DETECTED.
```

SYMBOL	VALUE	ATTR.	LOCN	LINK
IRQV	FFF8	84	2AA2	
SWIV	FFFA	84	2AAB	
NMIV	FFFC	84	2AB4	
RSTV	FFFE	84	2ABD	
AADD	FCD0	84	2AC6	
BADD	FCD2	84	2ACF	
CADD	FCD4	84	2AD8	
FREQ	0001	84	2AE1	
IQAD	0002	84	2AEA	
VDAD	0003	84	2AF3	
VQAD	0004	84	2AFC	
IDAD	0005	84	2B05	
ERMT	0012	84	2B0E	
ERLT	0013	84	2B17	
LLS	C000	84	2B20	
LMS	8000	84	2B29	
MPRM	0012	84	2B32	
MPRL	0013	84	2B3B	
MPDM	0010	84	2B44	
MPDL	0011	84	2B4D	
ADDL	0011	84	2B56	
ADDM	0010	84	2B5F	
SUML	0013	84	2B68	
SUMM	0012	84	2B71	
RADD	00FD	84	2B7A	
KADD	00FE	84	2B83	
PROD	0010	84	2B8C	
DIFF	0010	84	2B95	
ESTAR	0032	84	2B9E	
AXMS	0021	84	2BA7	
BUN1M	0026	84	2BB0	
BUN1LM	0028	84	2BB9	
CUNM	002A	84	2BC2	
CUNLM	002C	84	2BCB	
UN1	0032	84	2BD4	
XN	0047	84	2BDD	
UN	0034	84	2BE6	
CMB	002A	84	2BEF	
XN1T	0026	84	2BF8	
OLDF	0040	84	2C01	
NEWF	0041	84	2C0A	
TMCNT	0043	84	2C13	
ISTART	F800	84	2C1C	
PAS	F803	84	2C25	
PAS1	F808	84	2C2E	
PBS	F81B	84	2C37	
EVS	F82F	84	2C40	
INP	F85F	84	2C49	
INPUT	F840	84	2C52	
RECTFY	FB41	84	2C5B	
FLTCL	F9CA	84	2C64	
ERRST	FAD7	84	2C6D	
FZFG	0055	84	2C76	
CNFI	F874	84	2C7F	

SYMBOL	VALUE	ATTR	LOCN	LINK
CNFI1	F877	84	2C88	
GT6	F87F	84	2C91	
ADLOOP	F892	84	2C9A	
FLXCL	F8D0	84	2CA3	
AB	F8D6	84	2CAC	
PIQ	F8E0	84	2CB5	
MPY16	FB58	84	2CBE	
AC	F8E4	84	2CC7	
SCP1	F8E9	84	2CD0	
CVD	F903	84	2CD9	
ADD16	FB8D	84	2CE2	
AD	F906	84	2CEB	
SIQ	F90D	84	2CF4	
AE	F916	84	2CFD	
AF	F924	84	2D06	
PID	F92E	84	2D0F	
SCP3	F937	84	2D18	
CVQ	F94F	84	2D21	
SID	F959	84	2D2A	
AK	F960	84	2D33	
AL	F968	84	2D3C	
AM	F974	84	2D45	
WXCOR	F978	84	2D4E	
SCWX	F98A	84	2D57	
SCWX1	F98C	84	2D60	
NOCLMP	F9A7	84	2D69	
CXP	F9D7	84	2D72	
PXN	F9EC	84	2D7B	
CAXN	F9F2	84	2D84	
CMBT	0050	84	2D8D	
SPCMB	FA60	84	2D96	
STXN1	FA7C	84	2D9F	
CXSCL	FA82	84	2DA8	
PUN1	FABA	84	2DB1	
ESOTM	003A	84	2DBA	
ESOTL	003B	84	2DC3	
NFCHG	FAE7	84	2DCC	
SFTC	FAE0	84	2DD5	
STCNT	FAE4	84	2DDE	
BYPASS	FB30	84	2DE7	
SCALE	FB0C	84	2DF0	
CES	FB33	84	2DF9	
POS	FB46	84	2E02	
GAIN1	FB4E	84	2E0B	
MPIER	0014	84	2E14	
SMPY2	FB6F	84	2E1D	
SMPY1	FB67	84	2E26	
SMPY3	FB71	84	2E2F	
SMPY4	FB88	84	2E38	
EOUT	FB96	C4	2E41	

END ASM V1.1

```
0001 REM PROGRAM GENERATES VALUES FOR V PEAK NOM / V PEAK
0002 REM THIS IS TO NORMALIZE AFFECT OF B+ VARIATIONS
0003 REM ON THE OUTPUT VOLTAGE.
0004 REM
0005 LINE= 0
0010 PRINT #7,"    NAM V PEAK TABLE"
0020 PRINT #7,"     ORG $E500 "
0030 PRINT #7,"* "
0040 PRINT #7,"*  SCALING OF VNOM/VPEAK * 256 "
0050 PRINT #7,"*     GIVES SAME SCALING AS SCALE FACTOR"
0060 FOR B=0 TO 80 STEP 1
0070  B1=(128/80)*256
0080  B1=INT(B1)
0090 GOSUB 500
0100 NEXT B
0110 FOR B=81 TO 150 STEP 1
0120  B1=(128/B)*256
0130  B1=INT(B1)
0140 GOSUB 500
0150 NEXT B
0160 FOR B=151 TO 255 STEP 1
0170  B1=(128/150)*256
0180  B1=INT(B1)
0190 GOSUB 500
0200 NEXT B
0210 GOTO 800
0500 PRINT #4,"VP=",B,"128/V PEAK * 256=";B1
0501  K=B
0502 IF B<80 THEN K=80
0503 IF B>150 THEN K=150
0510 PRINT #7,"VP";B,"FDB   ";B1,";FACTOR=";128/K
0520 RETURN
0800 PRINT #7,"   END"
0810 END
```

```
SEQ  LOC  OBJ      SOURCE

0001                          NAM    V PEAK TABLE
0002           E500           ORG    $E500
0003                 *
0004                 *  SCALING OF VNOM/VPEAK * 256
0005                 *    GIVES SAME SCALING AS SCALE FACTOR
0006 E500 02 00     VP0       FDB    512            ;FACTOR =2
0007 E502 02 00     VP1       FDB    512            ;FACTOR =2
0008 E504 02 00     VP2       FDB    512            ;FACTOR =2
0009 E506 02 00     VP3       FDB    512            ;FACTOR =2
0010 E508 02 00     VP4       FDB    512            ;FACTOR =2
0011 E50A 02 00     VP5       FDB    512            ;FACTOR =2
0012 E50C 02 00     VP6       FDB    512            ;FACTOR =2
0013 E50E 02 00     VP7       FDB    512            ;FACTOR =2
0014 E510 02 00     VP8       FDB    512            ;FACTOR =2
0015 E512 02 00     VP9       FDB    512            ;FACTOR =2
0016 E514 02 00     VP10      FDB    512            ;FACTOR =2
0017 E516 02 00     VP11      FDB    512            ;FACTOR =2
0018 E518 02 00     VP12      FDB    512            ;FACTOR =2
0019 E51A 02 00     VP13      FDB    512            ;FACTOR =2
0020 E51C 02 00     VP14      FDB    512            ;FACTOR =2
0021 E51E 02 00     VP15      FDB    512            ;FACTOR =2
0022 E520 02 00     VP16      FDB    512            ;FACTOR =2
0023 E522 02 00     VP17      FDB    512            ;FACTOR =2
0024 E524 02 00     VP18      FDB    512            ;FACTOR =2
0025 E526 02 00     VP19      FDB    512            ;FACTOR =2
0026 E528 02 00     VP20      FDB    512            ;FACTOR =2
0027 E52A 02 00     VP21      FDB    512            ;FACTOR =2
0028 E52C 02 00     VP22      FDB    512            ;FACTOR =2
0029 E52E 02 00     VP23      FDB    512            ;FACTOR =2
0030 E530 02 00     VP24      FDB    512            ;FACTOR =2
0031 E532 02 00     VP25      FDB    512            ;FACTOR =2
0032 E534 02 00     VP26      FDB    512            ;FACTOR =2
0033 E536 02 00     VP27      FDB    512            ;FACTOR =2
0034 E538 02 00     VP28      FDB    512            ;FACTOR =2
0035 E53A 02 00     VP29      FDB    512            ;FACTOR =2
0036 E53C 02 00     VP30      FDB    512            ;FACTOR =2
0037 E53E 02 00     VP31      FDB    512            ;FACTOR =2
0038 E540 02 00     VP32      FDB    512            ;FACTOR =2
0039 E542 02 00     VP33      FDB    512            ;FACTOR =2
0040 E544 02 00     VP34      FDB    512            ;FACTOR =2
0041 E546 02 00     VP35      FDB    512            ;FACTOR =2
0042 E548 02 00     VP36      FDB    512            ;FACTOR =2
0043 E54A 02 00     VP37      FDB    512            ;FACTOR =2
0044 E54C 02 00     VP38      FDB    512            ;FACTOR =2
0045 E54E 02 00     VP39      FDB    512            ;FACTOR =2
0046 E550 02 00     VP40      FDB    512            ;FACTOR =2
0047 E552 02 00     VP41      FDB    512            ;FACTOR =2
0048 E554 02 00     VP42      FDB    512            ;FACTOR =2
0049 E556 02 00     VP43      FDB    512            ;FACTOR =2
0050 E558 02 00     VP44      FDB    512            ;FACTOR =2
0051 E55A 02 00     VP45      FDB    512            ;FACTOR =2
0052 E55C 02 00     VP46      FDB    512            ;FACTOR =2
0053 E55E 02 00     VP47      FDB    512            ;FACTOR =2
0054 E560 02 00     VP48      FDB    512            ;FACTOR =2
0055 E562 02 00     VP49      FDB    512            ;FACTOR =2
```

```
SEQ  LOC  OBJ        SOURCE

0056 E564 02 00      VP50      FDB      512          ;FACTOR =2
0057 E566 02 00      VP51      FDB      512          ;FACTOR =2
0058 E568 02 00      VP52      FDB      512          ;FACTOR =2
0059 E56A 02 00      VP53      FDB      512          ;FACTOR =2
0060 E56C 02 00      VP54      FDB      512          ;FACTOR =2
0061 E56E 02 00      VP55      FDB      512          ;FACTOR =2
0062 E570 02 00      VP56      FDB      512          ;FACTOR =2
0063 E572 02 00      VP57      FDB      512          ;FACTOR =2
0064 E574 02 00      VP58      FDB      512          ;FACTOR =2
0065 E576 02 00      VP59      FDB      512          ;FACTOR =2
0066 E578 02 00      VP60      FDB      512          ;FACTOR =2
0067 E57A 02 00      VP61      FDB      512          ;FACTOR =2
0068 E57C 02 00      VP62      FDB      512          ;FACTOR =2
0069 E57E 02 00      VP63      FDB      512          ;FACTOR =2
0070 E580 02 00      VP64      FDB      512          ;FACTOR =2
0071 E582 02 00      VP65      FDB      512          ;FACTOR =2
0072 E584 02 00      VP66      FDB      512          ;FACTOR =2
0073 E586 02 00      VP67      FDB      512          ;FACTOR =2
0074 E588 02 00      VP68      FDB      512          ;FACTOR =2
0075 E58A 02 00      VP69      FDB      512          ;FACTOR =2
0076 E58C 02 00      VP70      FDB      512          ;FACTOR =2
0077 E58E 02 00      VP71      FDB      512          ;FACTOR =2
0078 E590 02 00      VP72      FDB      512          ;FACTOR =2
0079 E592 02 00      VP73      FDB      512          ;FACTOR =2
0080 E594 02 00      VP74      FDB      512          ;FACTOR =2
0081 E596 02 00      VP75      FDB      512          ;FACTOR =2
0082 E598 02 00      VP76      FDB      512          ;FACTOR =2
0083 E59A 02 00      VP77      FDB      512          ;FACTOR =2
0084 E59C 02 00      VP78      FDB      512          ;FACTOR =2
0085 E59E 02 00      VP79      FDB      512          ;FACTOR =2
0086 E5A0 02 00      VP80      FDB      512          ;FACTOR =2
0087 E5A2 01 F9      VP81      FDB      505          ;FACTOR =1.97530864
0088 E5A4 01 F3      VP82      FDB      499          ;FACTOR =1.95121951
0089 E5A6 01 ED      VP83      FDB      493          ;FACTOR =1.92771084
0090 E5A8 01 E7      VP84      FDB      487          ;FACTOR =1.9047619
0091 E5AA 01 E1      VP85      FDB      481          ;FACTOR =1.88235294
0092 E5AC 01 DC      VP86      FDB      476          ;FACTOR =1.86046511
0093 E5AE 01 D6      VP87      FDB      470          ;FACTOR =1.83908045
0094 E5B0 01 D1      VP88      FDB      465          ;FACTOR =1.81818181
0095 E5B2 01 CC      VP89      FDB      460          ;FACTOR =1.7977528
0096 E5B4 01 C7      VP90      FDB      455          ;FACTOR =1.77777777
0097 E5B6 01 C2      VP91      FDB      450          ;FACTOR =1.75824175
0098 E5B8 01 BD      VP92      FDB      445          ;FACTOR =1.73913043
0099 E5BA 01 B8      VP93      FDB      440          ;FACTOR =1.7204301
0100 E5BC 01 B3      VP94      FDB      435          ;FACTOR =1.70212765
0101 E5BE 01 AF      VP95      FDB      431          ;FACTOR =1.68421052
0102 E5C0 01 AA      VP96      FDB      426          ;FACTOR =1.66666666
0103 E5C2 01 A6      VP97      FDB      422          ;FACTOR =1.64948453
0104 E5C4 01 A1      VP98      FDB      417          ;FACTOR =1.63265306
0105 E5C6 01 9D      VP99      FDB      413          ;FACTOR =1.61616161
0106 E5C8 01 99      VP100     FDB      409          ;FACTOR =1.6
0107 E5CA 01 95      VP101     FDB      405          ;FACTOR =1.58415841
0108 E5CC 01 91      VP102     FDB      401          ;FACTOR =1.56862745
0109 E5CE 01 8D      VP103     FDB      397          ;FACTOR =1.55339805
0110 E5D0 01 89      VP104     FDB      393          ;FACTOR =1.53846153
```

```
SEQ  LOC  OBJ      SOURCE

0111 E5D2 01 86    VP105    FDB     390          ;FACTOR =1.52380952
0112 E5D4 01 82    VP106    FDB     386          ;FACTOR =1.50943396
0113 E5D6 01 7E    VP107    FDB     382          ;FACTOR =1.4953271
0114 E5D8 01 7B    VP108    FDB     379          ;FACTOR =1.48148148
0115 E5DA 01 77    VP109    FDB     375          ;FACTOR =1.4678899
0116 E5DC 01 74    VP110    FDB     372          ;FACTOR =1.45454545
0117 E5DE 01 71    VP111    FDB     369          ;FACTOR =1.44144144
0118 E5E0 01 6D    VP112    FDB     365          ;FACTOR =1.42857142
0119 E5E2 01 6A    VP113    FDB     362          ;FACTOR =1.4159292
0120 E5E4 01 67    VP114    FDB     359          ;FACTOR =1.40350877
0121 E5E6 01 64    VP115    FDB     356          ;FACTOR =1.39130434
0122 E5E8 01 61    VP116    FDB     353          ;FACTOR =1.37931034
0123 E5EA 01 5E    VP117    FDB     350          ;FACTOR =1.36752136
0124 E5EC 01 5B    VP118    FDB     347          ;FACTOR =1.3559322
0125 E5EE 01 58    VP119    FDB     344          ;FACTOR =1.34453781
0126 E5F0 01 55    VP120    FDB     341          ;FACTOR =1.33333333
0127 E5F2 01 52    VP121    FDB     338          ;FACTOR =1.32231404
0128 E5F4 01 4F    VP122    FDB     335          ;FACTOR =1.3114754
0129 E5F6 01 4D    VP123    FDB     333          ;FACTOR =1.300813
0130 E5F8 01 4A    VP124    FDB     330          ;FACTOR =1.29032258
0131 E5FA 01 47    VP125    FDB     327          ;FACTOR =1.28
0132 E5FC 01 45    VP126    FDB     325          ;FACTOR =1.26984126
0133 E5FE 01 42    VP127    FDB     322          ;FACTOR =1.25984251
0134 E600 01 40    VP128    FDB     320          ;FACTOR =1.25
0135 E602 01 3D    VP129    FDB     317          ;FACTOR =1.24031007
0136 E604 01 3B    VP130    FDB     315          ;FACTOR =1.23076923
0137 E606 01 38    VP131    FDB     312          ;FACTOR =1.22137404
0138 E608 01 36    VP132    FDB     310          ;FACTOR =1.21212121
0139 E60A 01 33    VP133    FDB     307          ;FACTOR =1.20300751
0140 E60C 01 31    VP134    FDB     305          ;FACTOR =1.19402985
0141 E60E 01 2F    VP135    FDB     303          ;FACTOR =1.18518518
0142 E610 01 2D    VP136    FDB     301          ;FACTOR =1.17647058
0143 E612 01 2A    VP137    FDB     298          ;FACTOR =1.16788321
0144 E614 01 28    VP138    FDB     296          ;FACTOR =1.15942028
0145 E616 01 26    VP139    FDB     294          ;FACTOR =1.15107913
0146 E618 01 24    VP140    FDB     292          ;FACTOR =1.14285714
0147 E61A 01 22    VP141    FDB     290          ;FACTOR =1.13475177
0148 E61C 01 20    VP142    FDB     288          ;FACTOR =1.12676056
0149 E61E 01 1E    VP143    FDB     286          ;FACTOR =1.11888111
0150 E620 01 1C    VP144    FDB     284          ;FACTOR =1.11111111
0151 E622 01 1A    VP145    FDB     282          ;FACTOR =1.10344827
0152 E624 01 18    VP146    FDB     280          ;FACTOR =1.09589041
0153 E626 01 16    VP147    FDB     278          ;FACTOR =1.08843537
0154 E628 01 14    VP148    FDB     276          ;FACTOR =1.08108108
0155 E62A 01 12    VP149    FDB     274          ;FACTOR =1.0738255
0156 E62C 01 11    VP150    FDB     273          ;FACTOR =1.06666666
0157 E62E 01 0F    VP151    FDB     271          ;FACTOR =1.05960264
0158 E630 01 0D    VP152    FDB     269          ;FACTOR =1.05263157
0159 E632 01 0B    VP153    FDB     267          ;FACTOR =1.04575163
0160 E634 01 09    VP154    FDB     265          ;FACTOR =1.03896103
0161 E636 01 08    VP155    FDB     264          ;FACTOR =1.03225806
0162 E638 01 06    VP156    FDB     262          ;FACTOR =1.02564102
0163 E63A 01 04    VP157    FDB     260          ;FACTOR =1.01910828
0164 E63C 01 03    VP158    FDB     259          ;FACTOR =1.01265822
0165 E63E 01 01    VP159    FDB     257          ;FACTOR =1.0062893
```

```
SEQ  LOC  OBJ     SOURCE

0166 E640 01 00   VP160   FDB     256          ;FACTOR =1
0167 E642 00 FE   VP161   FDB     254          ;FACTOR =0.993788819
0168 E644 00 FC   VP162   FDB     252          ;FACTOR =0.98765432
0169 E646 00 FB   VP163   FDB     251          ;FACTOR =0.981595092
0170 E648 00 F9   VP164   FDB     249          ;FACTOR =0.975609756
0171 E64A 00 F8   VP165   FDB     248          ;FACTOR =0.969696969
0172 E64C 00 F6   VP166   FDB     246          ;FACTOR =0.963855421
0173 E64E 00 F5   VP167   FDB     245          ;FACTOR =0.958083832
0174 E650 00 F3   VP168   FDB     243          ;FACTOR =0.952380952
0175 E652 00 F2   VP169   FDB     242          ;FACTOR =0.946745562
0176 E654 00 F0   VP170   FDB     240          ;FACTOR =0.94117647
0177 E656 00 EF   VP171   FDB     239          ;FACTOR =0.935672514
0178 E658 00 EE   VP172   FDB     238          ;FACTOR =0.930232558
0179 E65A 00 EC   VP173   FDB     236          ;FACTOR =0.924855491
0180 E65C 00 EB   VP174   FDB     235          ;FACTOR =0.919540229
0181 E65E 00 EA   VP175   FDB     234          ;FACTOR =0.914285714
0182 E660 00 E8   VP176   FDB     232          ;FACTOR =0.909090909
0183 E662 00 E7   VP177   FDB     231          ;FACTOR =0.903954802
0184 E664 00 E6   VP178   FDB     230          ;FACTOR =0.898876404
0185 E666 00 E4   VP179   FDB     228          ;FACTOR =0.893854748
0186 E668 00 E3   VP180   FDB     227          ;FACTOR =0.888888888
0187 E66A 00 E2   VP181   FDB     226          ;FACTOR =0.8839779
0188 E66C 00 E1   VP182   FDB     225          ;FACTOR =0.879120879
0189 E66E 00 DF   VP183   FDB     223          ;FACTOR =0.874316939
0190 E670 00 DE   VP184   FDB     222          ;FACTOR =0.869565217
0191 E672 00 DD   VP185   FDB     221          ;FACTOR =0.864864864
0192 E674 00 DC   VP186   FDB     220          ;FACTOR =0.860215053
0193 E676 00 DB   VP187   FDB     219          ;FACTOR =0.855614973
0194 E678 00 D9   VP188   FDB     217          ;FACTOR =0.851063829
0195 E67A 00 D8   VP189   FDB     216          ;FACTOR =0.846560846
0196 E67C 00 D7   VP190   FDB     215          ;FACTOR =0.842105263
0197 E67E 00 D6   VP191   FDB     214          ;FACTOR =0.837696335
0198 E680 00 D5   VP192   FDB     213          ;FACTOR =0.833333333
0199 E682 00 D4   VP193   FDB     212          ;FACTOR =0.829015544
0200 E684 00 D3   VP194   FDB     211          ;FACTOR =0.824742268
0201 E686 00 D2   VP195   FDB     210          ;FACTOR =0.82051282
0202 E688 00 D0   VP196   FDB     208          ;FACTOR =0.81632653
0203 E68A 00 CF   VP197   FDB     207          ;FACTOR =0.812182741
0204 E68C 00 CE   VP198   FDB     206          ;FACTOR =0.808080808
0205 E68E 00 CD   VP199   FDB     205          ;FACTOR =0.8040201
0206 E690 00 CC   VP200   FDB     204          ;FACTOR =0.8
0207 E692 00 CC   VP201   FDB     204          ;FACTOR =0.8
0208 E694 00 CC   VP202   FDB     204          ;FACTOR =0.8
0209 E696 00 CC   VP203   FDB     204          ;FACTOR =0.8
0210 E698 00 CC   VP204   FDB     204          ;FACTOR =0.8
0211 E69A 00 CC   VP205   FDB     204          ;FACTOR =0.8
0212 E69C 00 CC   VP206   FDB     204          ;FACTOR =0.8
0213 E69E 00 CC   VP207   FDB     204          ;FACTOR =0.8
0214 E6A0 00 CC   VP208   FDB     204          ;FACTOR =0.8
0215 E6A2 00 CC   VP209   FDB     204          ;FACTOR =0.8
0216 E6A4 00 CC   VP210   FDB     204          ;FACTOR =0.8
0217 E6A6 00 CC   VP211   FDB     204          ;FACTOR =0.8
0218 E6A8 00 CC   VP212   FDB     204          ;FACTOR =0.8
0219 E6AA 00 CC   VP213   FDB     204          ;FACTOR =0.8
0220 E6AC 00 CC   VP214   FDB     204          ;FACTOR =0.8
```

```
SEQ  LOC  OBJ       SOURCE

0221 E6AE 00 CC     VP215    FDB     204            ;FACTOR =0.8
0222 E6B0 00 CC     VP216    FDB     204            ;FACTOR =0.8
0223 E6B2 00 CC     VP217    FDB     204            ;FACTOR =0.8
0224 E6B4 00 CC     VP218    FDB     204            ;FACTOR =0.8
0225 E6B6 00 CC     VP219    FDB     204            ;FACTOR =0.8
0226 E6B8 00 CC     VP220    FDB     204            ;FACTOR =0.8
0227 E6BA 00 CC     VP221    FDB     204            ;FACTOR =0.8
0228 E6BC 00 CC     VP222    FDB     204            ;FACTOR =0.8
0229 E6BE 00 CC     VP223    FDB     204            ;FACTOR =0.8
0230 E6C0 00 CC     VP224    FDB     204            ;FACTOR =0.8
0231 E6C2 00 CC     VP225    FDB     204            ;FACTOR =0.8
0232 E6C4 00 CC     VP226    FDB     204            ;FACTOR =0.8
0233 E6C6 00 CC     VP227    FDB     204            ;FACTOR =0.8
0234 E6C8 00 CC     VP228    FDB     204            ;FACTOR =0.8
0235 E6CA 00 CC     VP229    FDB     204            ;FACTOR =0.8
0236 E6CC 00 CC     VP230    FDB     204            ;FACTOR =0.8
0237 E6CE 00 CC     VP231    FDB     204            ;FACTOR =0.8
0238 E6D0 00 CC     VP232    FDB     204            ;FACTOR =0.8
0239 E6D2 00 CC     VP233    FDB     204            ;FACTOR =0.8
0240 E6D4 00 CC     VP234    FDB     204            ;FACTOR =0.8
0241 E6D6 00 CC     VP235    FDB     204            ;FACTOR =0.8
0242 E6D8 00 CC     VP236    FDB     204            ;FACTOR =0.8
0243 E6DA 00 CC     VP237    FDB     204            ;FACTOR =0.8
0244 E6DC 00 CC     VP238    FDB     204            ;FACTOR =0.8
0245 E6DE 00 CC     VP239    FDB     204            ;FACTOR =0.8
0246 E6E0 00 CC     VP240    FDB     204            ;FACTOR =0.8
0247 E6E2 00 CC     VP241    FDB     204            ;FACTOR =0.8
0248 E6E4 00 CC     VP242    FDB     204            ;FACTOR =0.8
0249 E6E6 00 CC     VP243    FDB     204            ;FACTOR =0.8
0250 E6E8 00 CC     VP244    FDB     204            ;FACTOR =0.8
0251 E6EA 00 CC     VP245    FDB     204            ;FACTOR =0.8
0252 E6EC 00 CC     VP246    FDB     204            ;FACTOR =0.8
0253 E6EE 00 CC     VP247    FDB     204            ;FACTOR =0.8
0254 E6F0 00 CC     VP248    FDB     204            ;FACTOR =0.8
0255 E6F2 00 CC     VP249    FDB     204            ;FACTOR =0.8
0256 E6F4 00 CC     VP250    FDB     204            ;FACTOR =0.8
0257 E6F6 00 CC     VP251    FDB     204            ;FACTOR =0.8
0258 E6F8 00 CC     VP252    FDB     204            ;FACTOR =0.8
0259 E6FA 00 CC     VP253    FDB     204            ;FACTOR =0.8
0260 E6FC 00 CC     VP254    FDB     204            ;FACTOR =0.8
0261 E6FE 00 CC     VP255    FDB     204            ;FACTOR =0.8
0262 E700 00 CC     VP256    FDB     204            ;FACTOR =0.8
0263                         END

0263 LINES ASSEMBLED,  LOC = E702,  0000 ERRORS DETECTED.

END ASM V1.1
```

```
0005 REM ****FTGN0****
0006 REM *** VALUES FROM VAC MEASUERED - UNLOADED ***
0010 REM PROGRAM CALCULATES & LISTS VALUES FOR FTBL
0020 REM THIS FOR NEW MODU SCHEME (EVERY P.P.-B+)
0030 REM F    = FREQ OF INTEREST
0040 REM N    = NO. OF PULSE PERIODS IN 60 DEGREES
0050 REM D(I) = DOMINANT PULSE LENGTH - TPDOM
0060 REM C(I) = 2ND COMPL PULSE LENGTH- TPCOM
0070 REM M1   = MAX SCALE FACTOR
0080 REM M2   = MIN SCALE FACTOR
0090 REM T    = PULSE PERIOD LENGTH   - TPP
0100 REM L    = LENGTH OF 60 DEG SEG
0110 REM B(F) = NO. OF P.P. IN 60 DEG - NPP
0120 REM
0125 REM "26 ON LINE 770 IS SHOOT THROUGH TIME"
0130 REM ESTABLISH NPP FOR EACH "F" (ALWAYS ODD)
0131 DIM B(60)
0132 INPUT "DISK OUTPUT ? YES=1,NO=0",Q1
0133 IF Q1=0 GOTO 1400
0140  B(6)=35
0150  B(7)=31
0160  B(8)=27
0170  B(9)=25
0180  B(10)=21
0190  B(11)=19
0200  B(12)=19
0210  B(13)=17
0220  B(14)=15
0230  B(15)=15
0240  B(16)=15
0250  B(17)=13
0260  B(18)=13
0270 FOR A1=19 TO 24
0280  B(A1)=11
0290 NEXT A1
0300 FOR A1=25 TO 29
0310  B(A1)=9
0320 NEXT A1
0330 FOR A1=30 TO 41
0340  B(A1)=7
0350 NEXT A1
0360 FOR A1=42 TO 55
0370  B(A1)=5
0380 NEXT A1
0390 FOR A1=56 TO 60
0400  B(A1)=3
0410 NEXT A1
0420 REM SET UP TOP OF DISK FILE TO BE 6800 FORMAT
0421 LINE= 0
0422 DIGITS= 0
0430 PRINT #7,"   NAM  FTBL"
0431 PRINT #4,"F";TAB(5);"VOPEN";TAB(20);"NPP";TAB(25);"TPP";
0432 PRINT #4,TAB(30);"L-60";TAB(40);"SFMAX"
0433 PRINT #7,"*"
0434 PRINT #7,"*GENERATED BY FTGN0*"
0435 GOTO 1200
0440 PRINT #7,"   ORG  $E880"
0450 REM NO. OF P.P. IS ESTABLISHED FOR EACH "F"
0460 REM NEXT DO DOMINANT AND COMPLEMENTARY
0470 REM PULSE WIDTHS FOR EACH P.P. IN "F"
0480  P1=3.14159
0490 DIM D(60),C(60),E(60),F(60)
0500 GOTO 1400
0510 IF Q1=0 GOTO 1085
```

```
0520  L1=1E+06/6/F
0530  T1=L1/N
0540  T=INT(T1+0.5)
0550  L=N*T
0600 FOR K=1 TO (N+1)/2 STEP 1
0610  K1=1/(2*P1*F)
0620  K2=(60/N)*(K-1)
0630  K3=(60/N)*K
0640  K7=(K2+60)*2*P1/360
0650  K8=(K3+60)*2*P1/360
0660  K9=(K2+120)*2*P1/360
0670  G1=(K3+120)*2*P1/360
0680  E(K)=K1*(COS(K7)-COS(K8))*1E+06+0.5
0690  F(K)=K1*(COS(K9)-COS(G1))*1E+06+0.5
0700 REM OLD TF*VOPEN/B*NOM= NEW TF
0710  D(K)=INT(E(K)*O/160)
0720  C(K)=INT(F(K)*O/160)
0730 NEXT K
0740 REM HAVE COMPLETED PULSE WIDTHS FOR "F".  DO SCALE FACTORS
0750  K4=(N+1)/2
0760  K5=D(K4)
0770  M=(T-26)/K5
0790  M1=INT(M*256+0.5)
0791 IF K5>(T-26) THEN M1=INT((T-26)*256/K5+0.5)
0792 IF M1>2047 THEN M1=2047
0795 PRINT #4,F,TAB(5);O;TAB(20);N;TAB(25);T;TAB(30);L;TAB(40);M1
0800 REM MIN SF = 0.50 ALWAYS
0810  M2=128
0820 REM OUTPUT VALUES TO TABLE ON DISK
0830 REM FTEL HAS FORM :
0840 REM N  = NPP
0850 REM T  = TPP
0860 REM L  = LENGTH OF 60 DEG (USEC)
0870 REM SFMAX
0880 REM SFMIN
0890 REM TPDOM #1 (USEC)
0900 REM TPCOM #1 (USEC)
0910 REM TPDOM #2
0920 REM TPCOM #2
0930 REM "
0940 REM "
0950 REM TPDOM (N+1)/2
0960 REM TPCOM (N+1)/2
0970 PRINT #7,"*FREQUENCY TABLE FOR ";F;"HZ."
0980 PRINT #7,"*"
0990 PRINT #7,"FRQ";F;" FCB ";N;"     ;NO.-PULSE PERIODS IN 60 DEG SEG"
1000 PRINT #7," FDB ",T,";LENGTH OF PULSE PERIOD (USEC)"
1010 PRINT #7," FDB ",L,";LENGTH OF 60 DEGREE SEG (USEC)"
1020 PRINT #7," FDB ",M1,";SFMAX"
1030 PRINT #7," FDB ",M2,";SFMIN"
1040 PRINT #7,"*PULSE WIDTHS"
1050 FOR I=1 TO (N+1)/2
1060 PRINT #7," ","FDB",D(I);"  ;DOM"
1070 PRINT #7," ","FDB",C(I)
1080 NEXT I
1081 RETURN
1085 PRINT #4,"F=";F,"O=";O
1086 RETURN
1100 IF Q1=0 GOTO 1800
1101 PRINT #7,"    END"
1110 GOTO 1800
1200 REM SECTION WRITES FREQ INDEX CHARACTERS FOR DISK FILE
1210 PRINT #7,"* * * * * * * * * * * * *"
1220 PRINT #7,"*FREQUENCY TABLES      *"
1230 PRINT #7,"* * * * * * * * * * * * *"
```

```
1250 PRINT #7, "*"
1255 PRINT #7, "INDEX", "FDB", "FRQ6"
1260 FOR I=7 TO 60
1270 PRINT #7, "  ", "FDB", "FRQ", I
1280 NEXT I
1290 GOTO 440
1400 FOR F=6 TO 60
1410  O=1.9*F
1420 GOSUB 510
1430 NEXT F
1440 GOTO 1100
1800 END
```

F=6	O=11.4
F=7	O=13.3
F=8	O=15.2
F=9	O=17.1
F=10	O=19
F=11	O=20.9
F=12	O=22.8
F=13	O=24.7
F=14	O=26.6
F=15	O=28.5
F=16	O=30.4
F=17	O=32.3
F=18	O=34.2
F=19	O=36.1
F=20	O=38
F=21	O=39.9
F=22	O=41.8
F=23	O=43.7
F=24	O=45.6
F=25	O=47.5
F=26	O=49.4
F=27	O=51.3
F=28	O=53.2
F=29	O=55.1
F=30	O=57
F=31	O=58.9
F=32	O=60.8
F=33	O=62.7
F=34	O=64.6
F=35	O=66.5
F=36	O=68.4
F=37	O=70.3
F=38	O=72.2
F=39	O=74.1
F=40	O=76
F=41	O=77.9
F=42	O=79.8
F=43	O=81.7
F=44	O=83.6
F=45	O=85.5
F=46	O=87.4
F=47	O=89.3
F=48	O=91.2
F=49	O=93.1
F=50	O=95
F=51	O=96.9
F=52	O=98.8
F=53	O=100.7
F=54	O=102.6
F=55	O=104.5
F=56	O=106.4
F=57	O=108.3
F=58	O=110.2
F=59	O=112.1
F=60	O=114

```
0005 REM **** VSRSG ****
0006 REM ***** THIS VERSION GENERATES LINEAR RELATIONSHIP
0007 REM FOR ALL VALUES. ACTUAL TABLE WAS MODIFIED BY HAND
0010 REM PROGRAM GENERATES REFERENCE VALUES
0020 REM FOR V* TABLE USED BY MPA TO DO
0030 REM OVERLOAD DETECTION.
0040 REM VALUES COME FROM MEASUREMENTS MADE
0050 REM ON MACHINE AT 117 V B+
0060 REM CREATE LEAD IN ON DISK
0070 PRINT #7," NAM VSREF"
0080 PRINT #7," ORG $E400"
0085 PRINT #4,"F","V* REF"
0090 FOR F=0 TO 10
0100  V=0
0110 GOSUB 1006
0120 NEXT F
0130 FOR F=11 TO 15
0140  V=(144-0)*(F-10)/5+0
0150 GOSUB 1000
0160 NEXT F
0170 FOR F=16 TO 18
0180  V=(336-144)*(F-15)/3+144
0190 GOSUB 1000
0200 NEXT F
0210 FOR F=19 TO 20
0220  V=(544-336)*(F-18)/2+336
0230 GOSUB 1000
0240 NEXT F
0250  F=21
0260  V=688
0270 GOSUB 1000
0290 FOR F=22 TO 24
0300  V=(1264-688)*(F-21)/3+688
0310 GOSUB 1000
0320 NEXT F
0330 FOR F=25 TO 26
0340  V=(1584-1264)*(F-24)/2+1264
0350 GOSUB 1000
0360 NEXT F
0370 FOR F=27 TO 29
0380  V=(1888-1584)*(F-26)/3+1584
0390 GOSUB 1000
0400 NEXT F
0410 FOR F=30 TO 31
0420  V=(2032-1888)*(F-29)/2+1888
0430 GOSUB 1000
0440 NEXT F
0450 FOR F=32 TO 37
0460  V=(2480-2032)*(F-31)/6+2032
0470 GOSUB 1000
0480 NEXT F
0490 FOR F=38 TO 55
0500  V=(3024-2480)*(F-37)/23+2480
0510 GOSUB 1000
0520 NEXT F
0530 FOR F=56 TO 60
0540  V=3583+(56-F)*(3583-3375)/(60-56)
0550 GOSUB 1006
0560 NEXT F
0570 GOTO 1500
1000 REM SUBROUTINE WRITES EACH LINE OF VSREF
1002 REM V* IN MPA AFTER SHIFTING IS 2 TIMES E*
1003 REM IN MPB AT FAF1
1005  V=1.25*V
```

```
1007  V1=16*V1+15
1008 IF F<11 GOTO 1010
1009 IF V1>V+8 THEN V1=V1-16
1010 PRINT #7,"VS";F;"    FDB ";V1;"    ,V*=";V
1020 PRINT #4,F,V
1030 RETURN
1500 PRINT #7,"  END"
1510 END
```

```
0005 REM **** R*GN ****
0010 REM PROGRAM MAKES TABLE OF R* VALUES
0015 REM VALUES OF R* ESTABLISHED BY MEASUREMENTS
0016 REM ON ACTUAL MACHINE
0020 PRINT #7, "  NAM  R* VALUES"
0030 PRINT #7, "  ORG $FD06"
0035 PRINT #7, "* VALUES ESTABLISHED BY MEASUREMENTS ON MACHINE"
0040 DIM R(60)
0050  R(6)=32
0055  R(7)=29
0060  R(8)=27
0065  R(9)=24
0070  R(10)=22
0075  R(11)=19
0080  R(12)=16
0085  R(13)=14
0090  R(14)=11
0095  R(15)=9
0100  R(16)=8
0105  R(17)=6
0110  R(18)=5
0115  R(19)=3
0120  R(20)=2
0125  R(21)=2
0130  R(22)=9
0135  R(23)=3
0140  R(24)=4
0145  R(25)=12
0150  R(26)=11
0155  R(27)=12
0160  R(28)=14
0165  R(29)=15
0170  R(30)=23
0175  R(31)=25
0180  R(32)=28
0185  R(33)=30
0190  R(34)=33
0195  R(35)=34
0200  R(36)=36
0205  R(37)=38
0210  R(38)=39
0215  R(39)=42
0220  R(40)=44
0225  R(41)=51
0230  R(42)=64
0235  R(43)=69
0240  R(44)=71
0245  R(45)=72
0250  R(46)=69
0255  R(47)=66
0260  R(48)=64
0265  R(49)=62
0270  R(50)=59
0275  R(51)=57
0280  R(52)=54
0285  R(53)=52
0290  R(54)=49
0295  R(55)=46
0300  R(56)=94
0305  R(57)=93
0310  R(58)=96
0315  R(59)=79
0320  R(60)=86
0330 FOR K=6 TO 60
```

```
0350 PRINT #4, "F="; K, "R*="; R(K)
0360 NEXT K
0370 PRINT #7, " END"
0380 END
```

```
0005 REM **** K1GN ****
0010 REM PROGRAM GENERATES K1(F) TABLE
0020 REM F=FREQUENCY
0030 REM K=DESIRED CONSTANT(2*PI*F*L1)
0040 REM L1=SERIES INDUCTANCE
0042 REM SCALING ON K HAS 13BITS AFTER DECIMAL
0050 INPUT "L1",L1
0060  P1=3.1415927
0062 GOSUB 200
0070 FOR F=6 TO 60 STEP 1
0080  K=2*P1*F*L1
0090 PRINT #4,F,K
0092 PRINT #7,"K",F,"FDB ";INT(K*8192);"    ;K=";K
0100 NEXT F
0110 PRINT #7,"  END"
0115 GOTO 800
0200 PRINT #4,"F","K1"
0210 PRINT #7,"  NAM K1 VALUES"
0220 PRINT #7,"  ORG $FE06"
0221 PRINT #7,"* TABLE GENERATED USING L1 =";L1
0230 RETURN
0800 END
```

[illegible]	FE00	01	2[illegible]	[illegible]	F[illegible]	[illegible]	.E+0 0[illegible]
[illegible]	FE0[illegible]	01	5[illegible]	[illegible]	[illegible]	1[illegible]	.E+0 0[illegible]
[illegible]	FE0[illegible]	01	[illegible]	[illegible]	F[illegible]	[illegible]	.E+0 0[illegible]
0007	FE0[illegible]	[illegible]	[illegible]	[illegible]	F[illegible]	[illegible]	.E+0 0[illegible]
[illegible]	FE0[illegible]	[illegible]	[illegible]	[illegible]	F[illegible]	[illegible]	.E+0 0[illegible]
[illegible]	FE[illegible]	02	[illegible]	[illegible]	F[illegible]	5[illegible]	.E+0 067[illegible]
[illegible]	FE[illegible]	02	[illegible]	[illegible]	F[illegible]	6[illegible]	.E+0 071[illegible]727
[illegible]	FE[illegible]	02	[illegible]	[illegible]	F[illegible]	601	.E+0 079[illegible]760121
[illegible]	FE[illegible]	02	[illegible]	[illegible]	F[illegible]	701	.E+0 0855895515
[illegible]	FE[illegible]	02	[illegible]	[illegible]	F[illegible]	701	.E+0 0917020909
[illegible]	FE14	0[illegible]	[illegible]	R1[illegible]	F[illegible]	801	.E+0 0978166299
[illegible]	FE1[illegible]	02	[illegible]	R17	F[illegible]	801	.E+0 10[illegible]
[illegible]	FE1[illegible]	02	[illegible]	R1[illegible]	F[illegible]	901	.E+0 1100[illegible]
[illegible]	FE20	0[illegible]	[illegible]	R1[illegible]	F[illegible]	901	.E+0 11615[illegible]
[illegible]	FE2[illegible]	[illegible]	[illegible]	R20	F[illegible]	1001	.E+0 1222[illegible]
[illegible]	FE2[illegible]	04	16	R21	F[illegible]	1001	.E+0 128[illegible]
[illegible]	FE2[illegible]	04	[illegible]	R2[illegible]	F[illegible]	1101	.E+0 1[illegible]
[illegible]	FE2[illegible]	04	7[illegible]	R2[illegible]	F[illegible]	1101	.E+0 1[illegible]6511405
[illegible]	FE2A	04	B[illegible]	R2[illegible]	[illegible]	1201	.E+0 1[illegible]
[illegible]	FE2C	04	[illegible]	R2[illegible]	F[illegible]	1[illegible]	.E+0 1[illegible]
[illegible]	FE2E	05	1[illegible]	R26	F[illegible]	1[illegible]	.E+0 1[illegible]
[illegible]	FE30	05	[illegible]	R27	F[illegible]	1[illegible]	.E+0 1[illegible]5062
[illegible]	FE32	05	7A	R28	F[illegible]	1[illegible]	.E+0 1[illegible]
[illegible]	FE34	05	[illegible]	R2[illegible]	F[illegible]	1[illegible]	.E+0 17[illegible]
[illegible]	FE36	05	[illegible]	R[illegible]	F[illegible]	1[illegible]	.E+0 18[illegible]
[illegible]	FE38	06	10	R[illegible]	F[illegible]	[illegible]	.E+0 1[illegible]
[illegible]	FE3A	06	[illegible]	R[illegible]	F[illegible]	1[illegible]	.E+0 1[illegible]
[illegible]	FE3C	06	[illegible]	R[illegible]	[illegible]	1[illegible]	.E+0 20[illegible]
[illegible]	FE3E	06	A[illegible]	R[illegible]	[illegible]	1701	.E+0 20[illegible]
[illegible]	FE[illegible]	06	[illegible]	R[illegible]	F[illegible]	1[illegible]	.E+0 21[illegible]
[illegible]	FE42	07	0A	R36	F[illegible]	1[illegible]	.E+0 22006[illegible]
[illegible]	FE4[illegible]	07	3D	R37	F[illegible]	1801	.E+0 2262[illegible]
[illegible]	FE46	07	6F	R38	F[illegible]	1[illegible]	.E+0 23[illegible]
[illegible]	FE4[illegible]	07	A1	R39	F[illegible]	1[illegible]	.E+0 2[illegible]5025
[illegible]	FE4A	07	D2	R40	F[illegible]	2001	.E+0 2[illegible]
[illegible]	FE4C	08	05	R41	F[illegible]	20[illegible]	.E+0 250[illegible]
[illegible]	FE4E	08	37	R42	F[illegible]	[illegible]	.E+0 25676[illegible]
[illegible]	FE5[illegible]	0[illegible]	[illegible]	R43	F[illegible]	[illegible]	.E+0 2[illegible]
[illegible]	FE5[illegible]	08	9[illegible]	R[illegible]	[illegible]	[illegible]	.E+0 2[illegible]
[illegible]	FE5[illegible]	08	[illegible]	R4[illegible]	F[illegible]	2[illegible]	.E+0 27[illegible]
[illegible]	FE56	08	FF	R46	F[illegible]	2[illegible]	.E+0 2[illegible]2811
[illegible]	FE58	09	31	R47	F[illegible]	[illegible]	.E+0 2[illegible]
[illegible]	FE5A	09	6[illegible]	R48	F[illegible]	[illegible]	.E+0 2[illegible]
[illegible]	FE5[illegible]	09	96	R49	F[illegible]	[illegible]	.E+0 2[illegible]
[illegible]	FE5E	09	C[illegible]	R50	F[illegible]	[illegible]	.E+0 3[illegible]
[illegible]	FE60	09	FA	R51	F[illegible]	[illegible]	.E+0 3117[illegible]
[illegible]	FE62	0A	2[illegible]	R52	F[illegible]	[illegible]	.E+0 3179[illegible]
[illegible]	FE6[illegible]	0A	5E	R5[illegible]	F[illegible]	[illegible]	.E+0 3[illegible]017[illegible]
[illegible]	FE6[illegible]	0A	[illegible]	R5[illegible]	F[illegible]	[illegible]	.E+0 3[illegible]
[illegible]	FE6[illegible]	0A	[illegible]	R5[illegible]	F[illegible]	[illegible]	.E+0 3[illegible]666
[illegible]	FE6[illegible]	0A	[illegible]	R5[illegible]	F[illegible]	28[illegible]	.E+0 3[illegible]58205
[illegible]	FE[illegible]	0[illegible]	[illegible]	R5[illegible]	F[illegible]	[illegible]	.E+0 3[illegible]

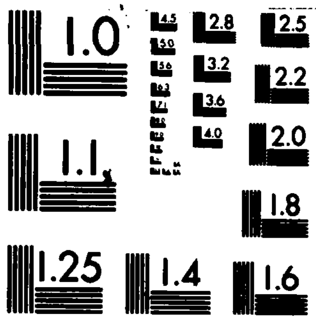
```
0010 DIGITS= 0
0015 LINE= 0
0050 REM PROGRAM CALCULATES FILTER CONSTANTS
0055 REM FORM OF EQUATION IS A*X-B*U1+C*U
0100 INPUT "W1-H, W2-L, DT", W1, W2, D
0130  A=EXP(-W2*D)
0135  B1=(W1-W2)/(D*W2*W2)
0136  B2=(1-W2*D-A)
0137  C1=1-(1+W2*D)*A
0140  B=B1*B2
0150  C=B1*C1
0160 PRINT "A=", A, "B=", B, "C=", C
0165 PRINT #4, "A="; A, "B="; B, "C="; C
0170  K=1/16384
0180  A1=A/K
0190  A2=B/K
0200  A3=C/K
0210 PRINT A1, A2, A3
0211 PRINT #4, A1, A2, A3
0220  X=INT(A1)
0221 PRINT X
0230 GOSUB 500
0240 PRINT #4, "A="; M4; M3; M2; M1
0250  X=INT(A2)
0251 PRINT X
0260 GOSUB 500
0270 PRINT #4, "B="; M4; M3; M2; M1
0280  X=INT(A3)
0281 PRINT X
0290 GOSUB 500
0300 PRINT #4, "C="; M4; M3; M2; M1
0310 GOTO 1000
0500  M4=INT(X/4096)
0510  X=INT(X-M4*4096)
0520  M3=INT(X/256)
0530  X=INT(X-M3*256)
0540  M2=INT(X/16)
0550  X=INT(X-M2*16)
0560  M1=X
0570 RETURN
1000 END
```

AD-A140 299 ADVANCED MOTOR-CONTROLLER DEVELOPMENT(U) WESTINGHOUSE OCEANIC DIV ANNAPOLIS MD L E LESSTER ET AL. 22 JUN 83 DTNSRDC-PASD-CR-1-83 N00167-82-C-0122

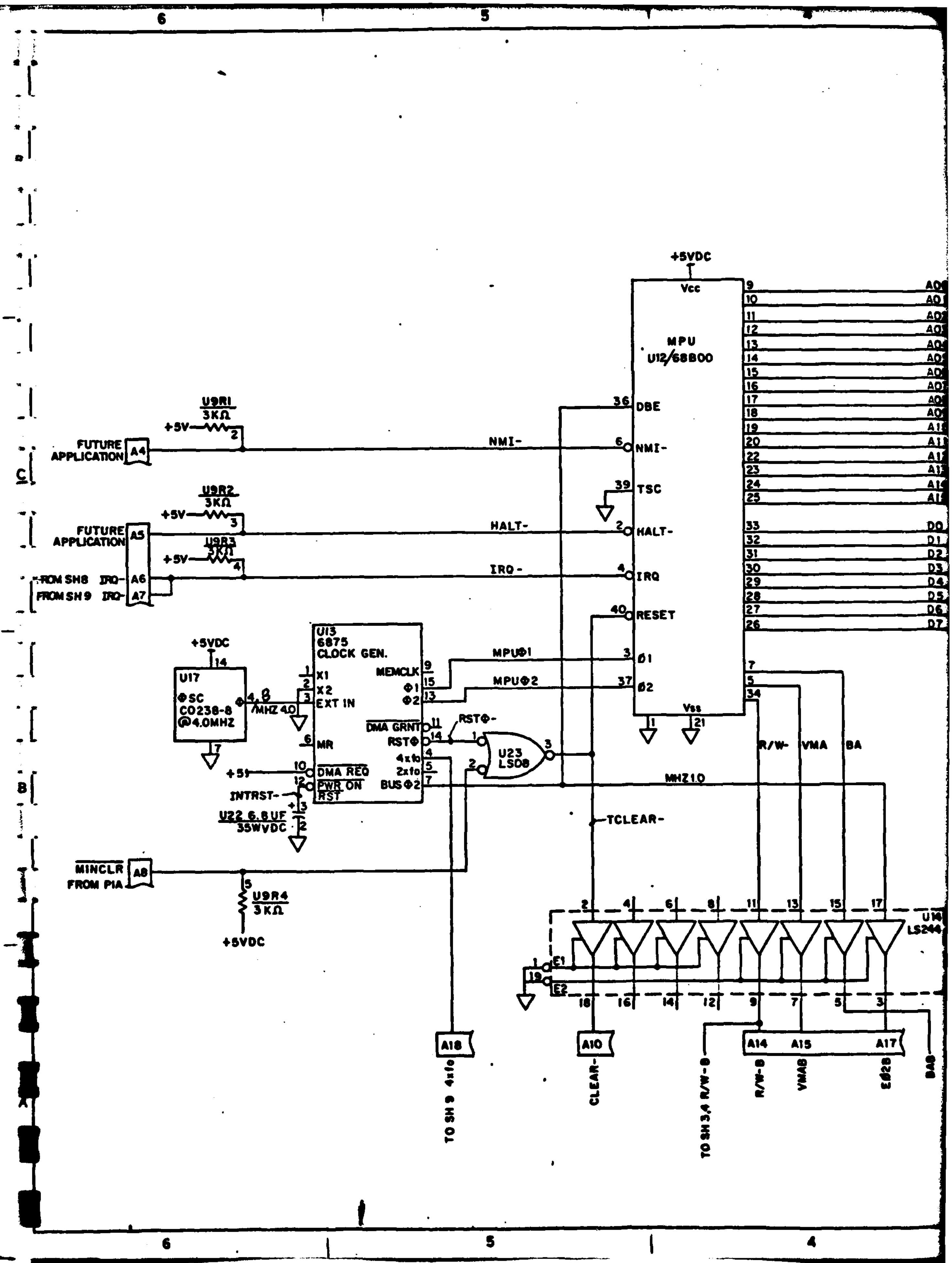
4/4

UNCLASSIFIED F/G 9/3 NL

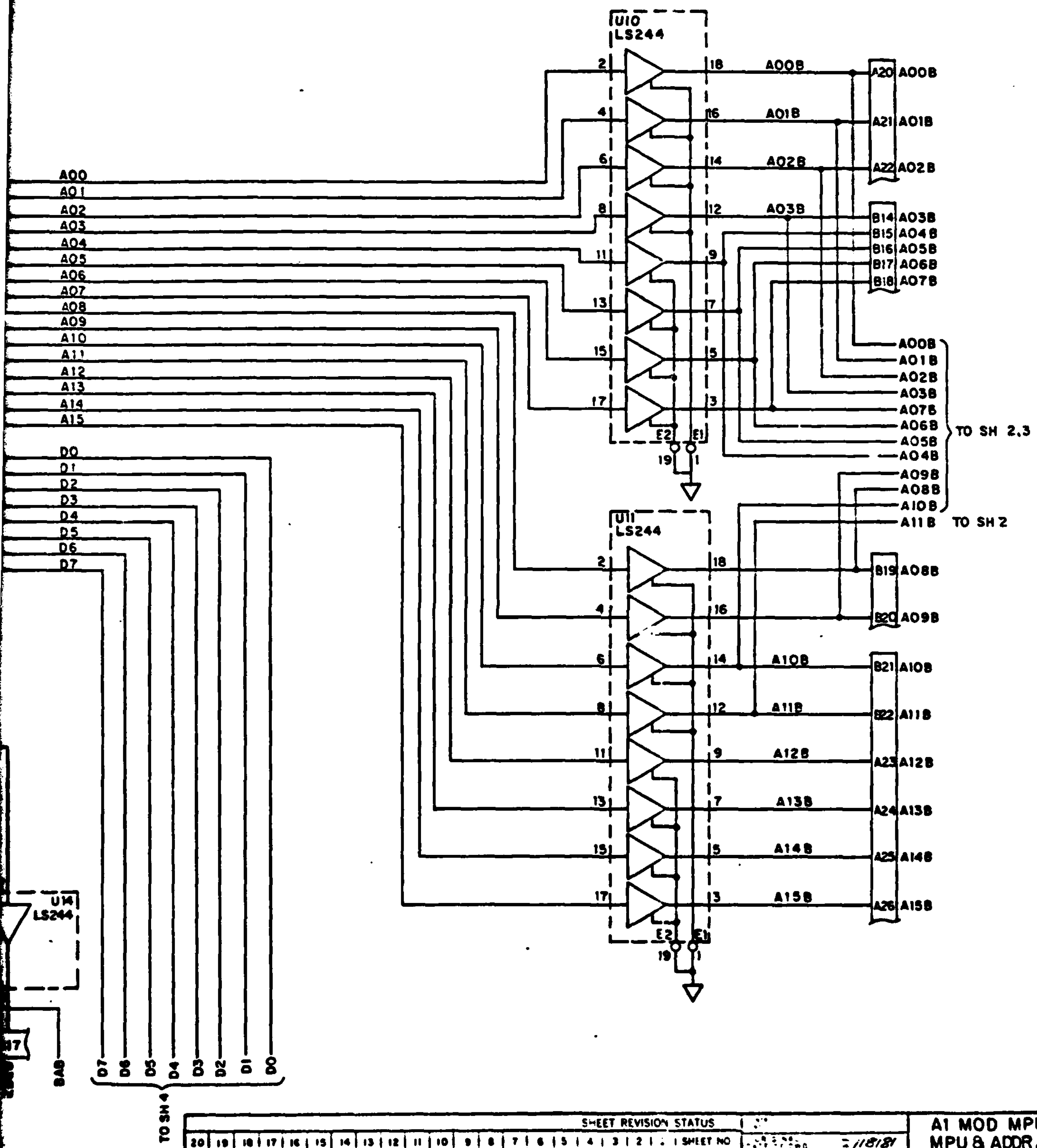
END
DATE
FILMED
6-84
DTIC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A



+5VDC
Vcc
MPU
U12/68B00
DBE
NMI-
TSC
HALT-
IRQ
RESET
Ø1
Ø2
Vss
U9R1
3KΩ
+5V
FUTURE
APPLICATION
A4
U9R2
3KΩ
A5
U9R3
3KΩ
FROM SH8 IRQ- A6
FROM SH9 IRQ- A7
NMI-
HALT-
IRQ -
U13
6875
CLOCK GEN.
X1
X2
EXT IN
MR
DMA REQ
PWR ON
RST
MEMCLK
Φ1
Φ2
DMA GRNT
RSTΦ
4xfo
2xfo
BUSΦ2
MPUΦ1
MPUΦ2
RSTΦ-
U23
LS08
MHZ1.0
U17
ΦSC
CO238-8
@4.0MHZ
4.0 MHZ
+5VDC
+5V
INTRST-
U22 6.8UF
35WVDC
MINCLR
FROM PIA
A8
U9R4
3KΩ
+5VDC
TCLEAR-
R/W-
VMA
BA
U14
LS244
E1
E2
A18
A10
A14
A15
A17
TO SH 9 4xfo
CLEAR-
TO SH 3,4 R/W-B
R/W-B
VMAB
EØ2B
BAB
A0
A1
D0
D1
D2
D3
D4
D5
D6
D7



SHEET REVISION STATUS

20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	SHEET NO
																				LATEST REV LTR

NEXT ASSY | USED ON

APPLICATION

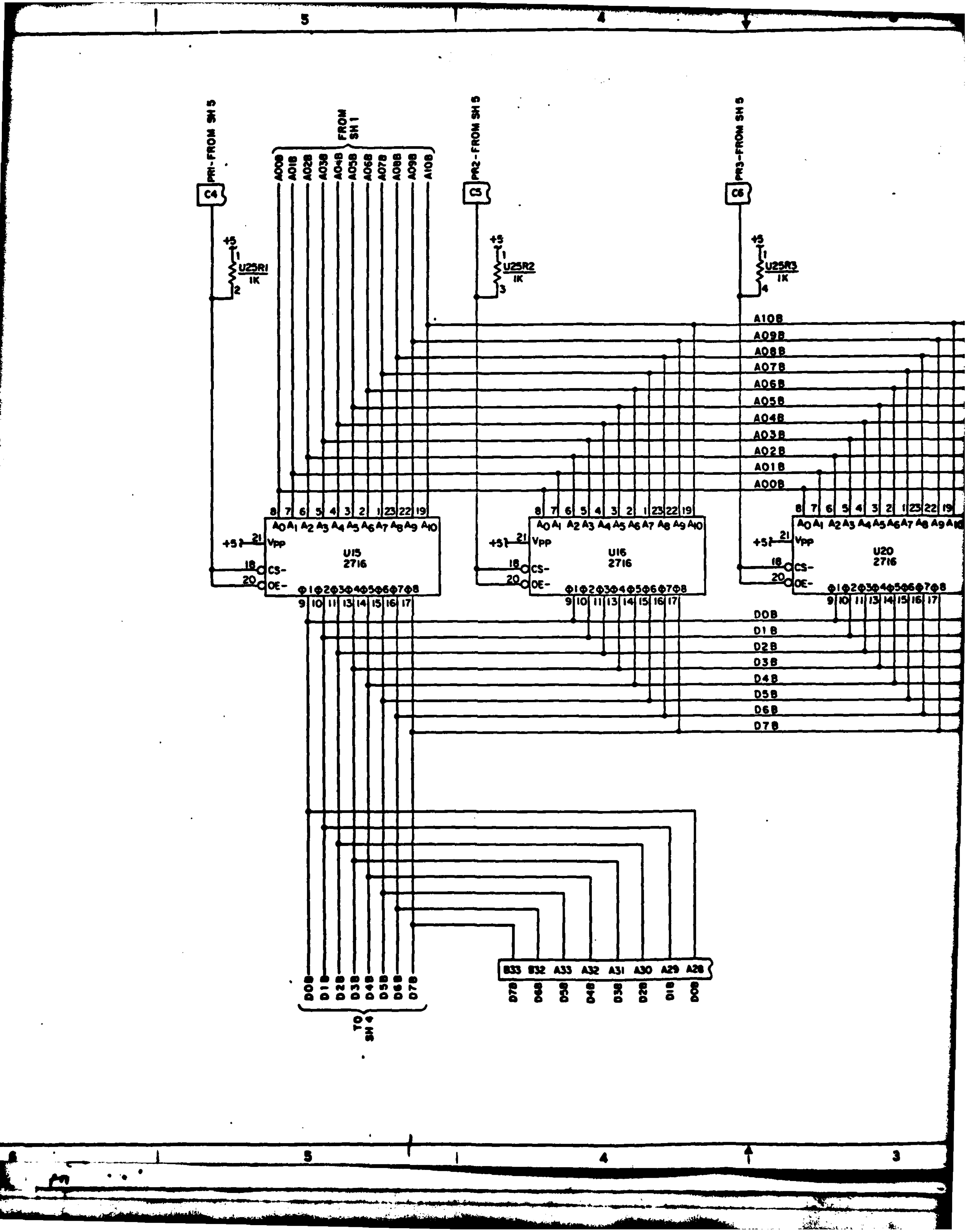
O PENNONI

2/18/81

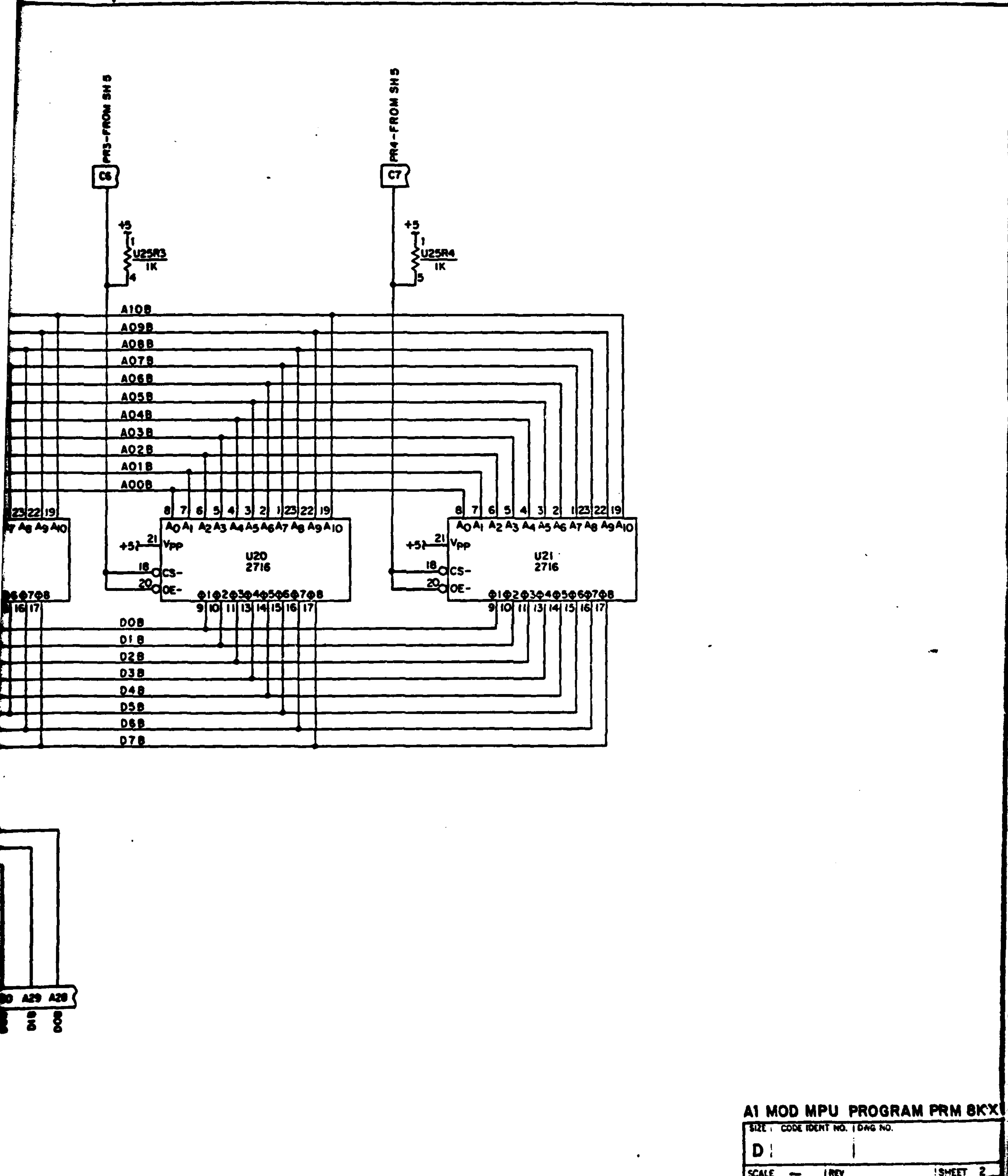
A1 MOD MPU
MPU & ADDR/CTRL BUFFERS

MICROPROCESSOR
LOGIC DIAGRAM

SIZE D

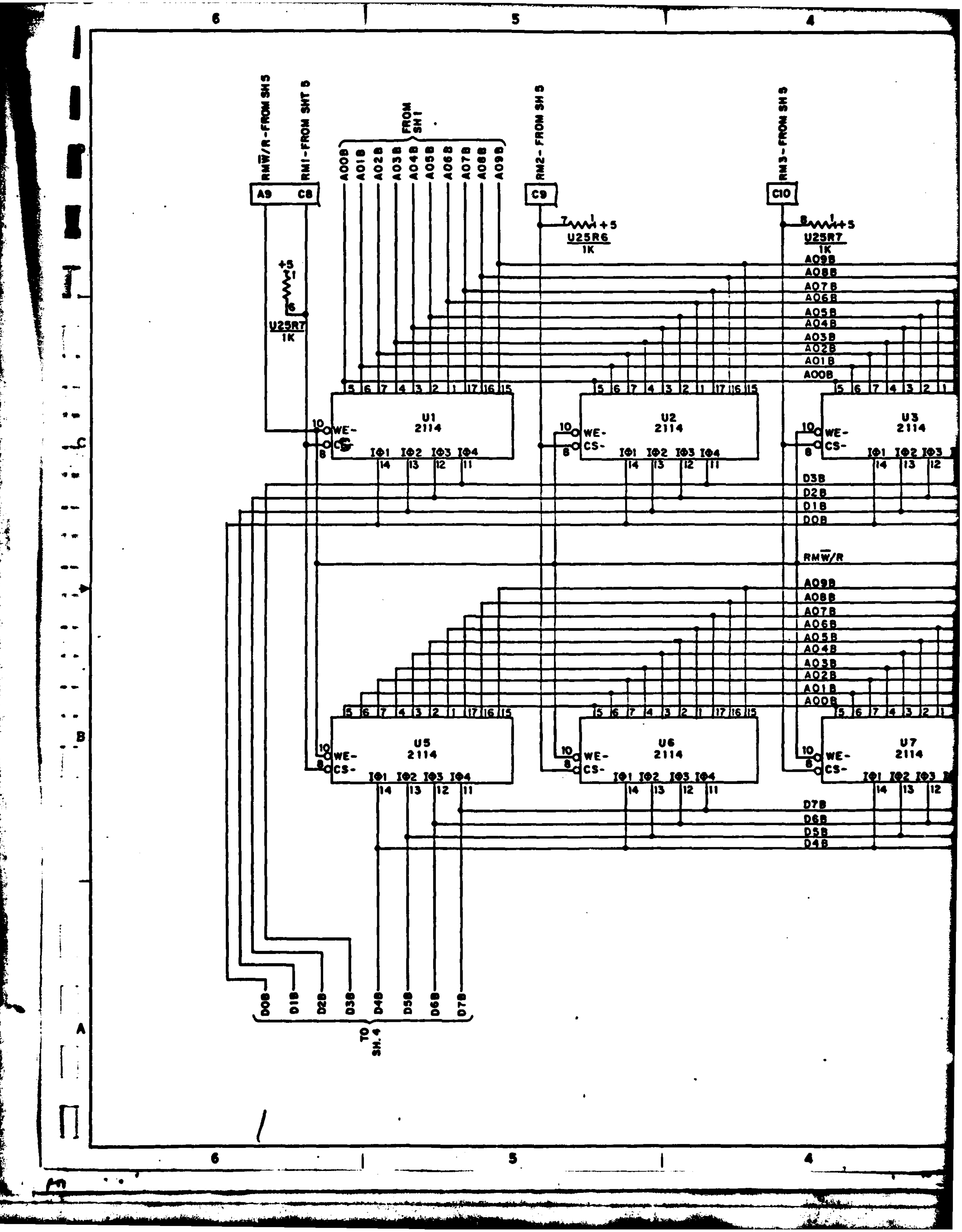


PR1-FROM SH 5
PR2-FROM SH 5
PR3-FROM SH 5
FROM SH 1
A00B A01B A02B A03B A04B A05B A06B A07B A08B A09B A10B
C4
C5
C6
+5
U25R1
1K
U25R2
1K
U25R3
1K
U15
2716
U16
2716
U20
2716
Vpp
CS-
OE-
D0B D1B D2B D3B D4B D5B D6B D7B
B33 B32 A33 A32 A31 A30 A29 A28
TO SH 4

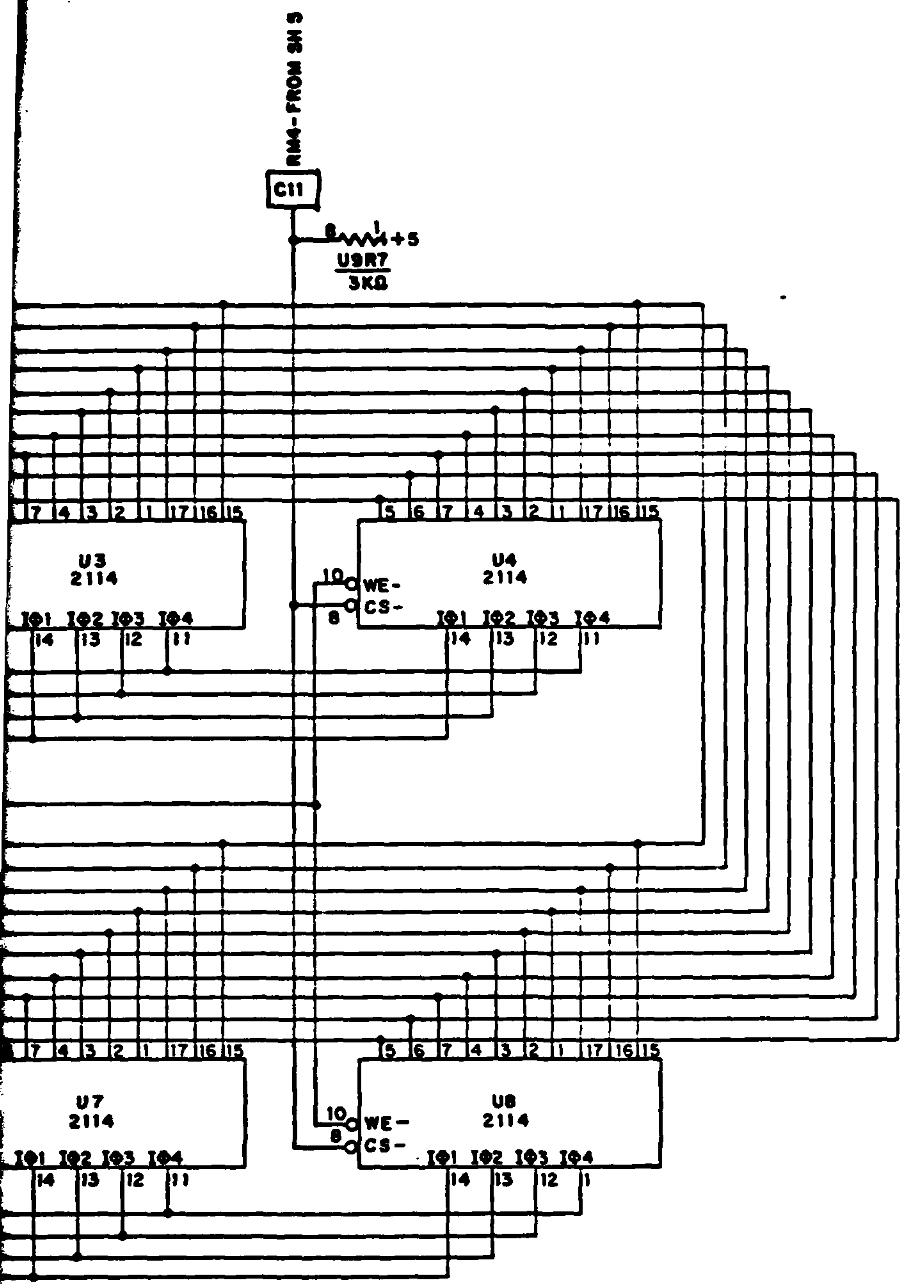


A1 MOD MPU PROGRAM PRM 8K X

SIZE	CODE IDENT NO.	DWG NO.
D		
SCALE —	REV	SHEET 2

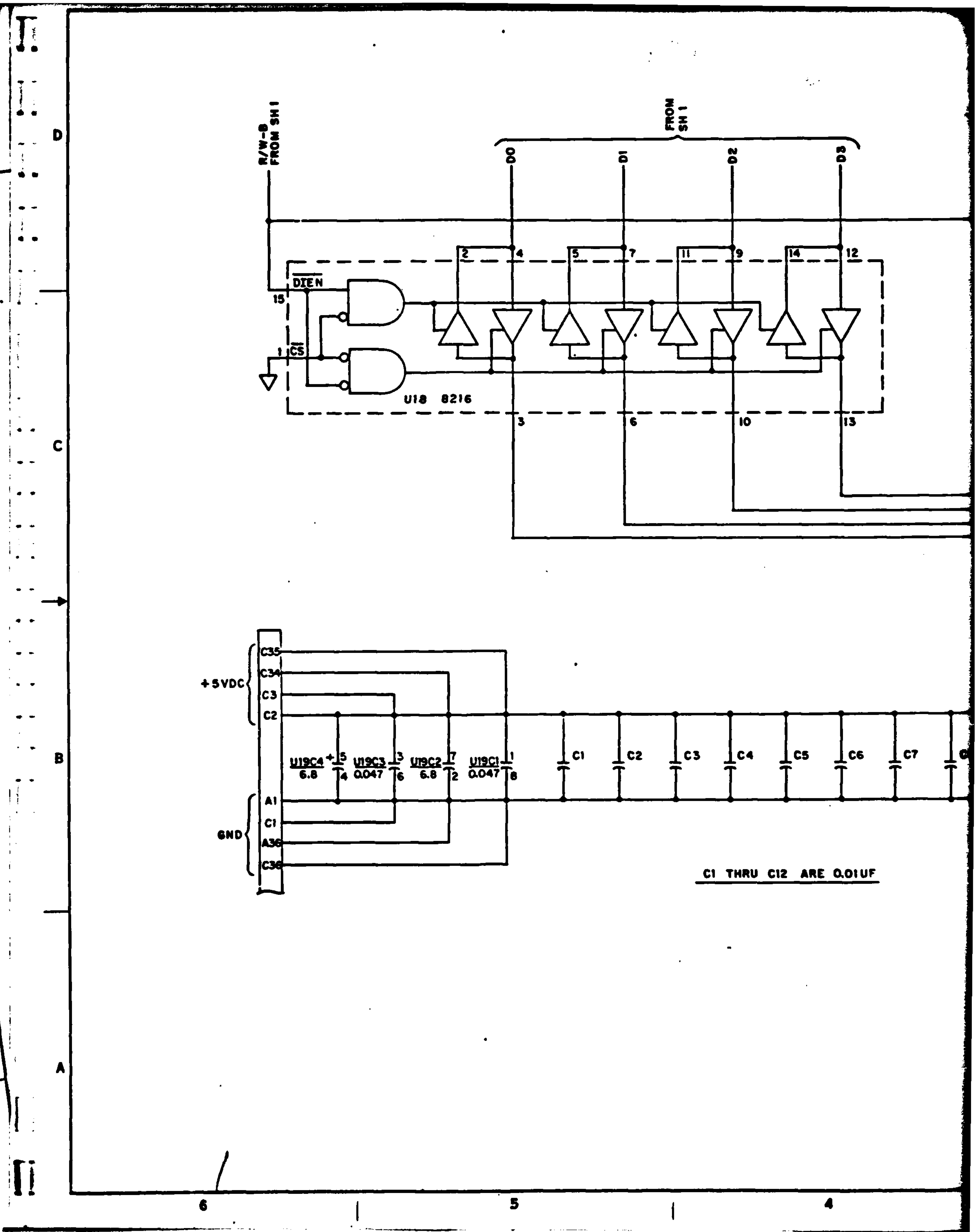


RMW/R-FROM SH5
RM1-FROM SHT 5
FROM SH1
A00B A01B A02B A03B A04B A05B A06B A07B A08B A09B
RM2-FROM SH5
RM3-FROM SH5
A9 C8
C9
C10
U25R6
1K
U25R7
1K
U1
2114
U2
2114
U3
2114
U5
2114
U6
2114
U7
2114
WE-
CS-
D3B D2B D1B D0B
RMW/R
D7B D6B D5B D4B
D0B D1B D2B D3B D4B D5B D6B D7B
TO SH.4

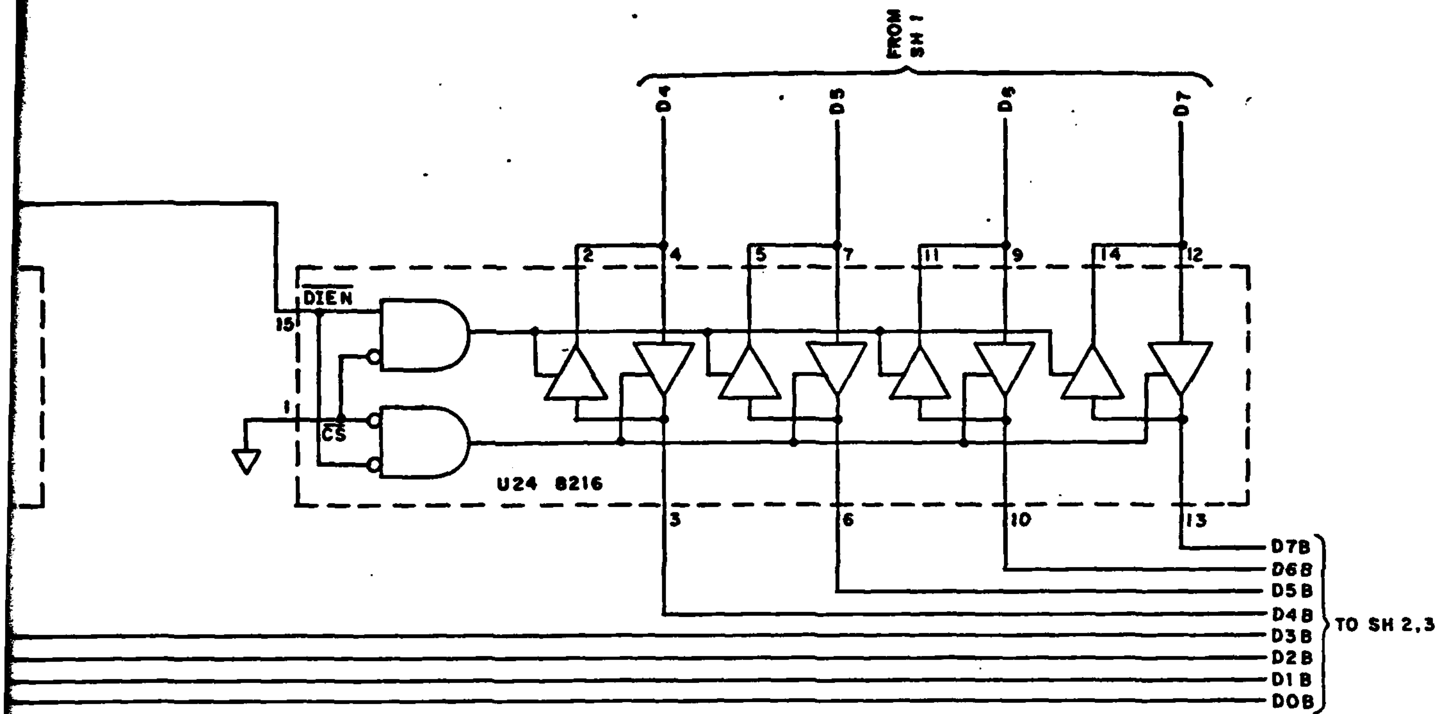


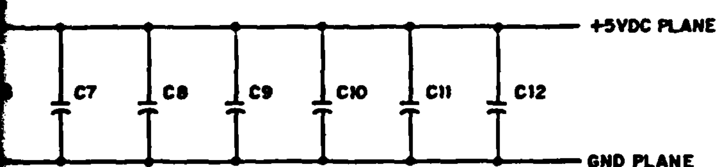
A1 MOD MPU PROGRAM RAM 4KX

SIZE	CODE IDENT NO.	DWG NO.
D		
SCALE —	REV	SHEET 3

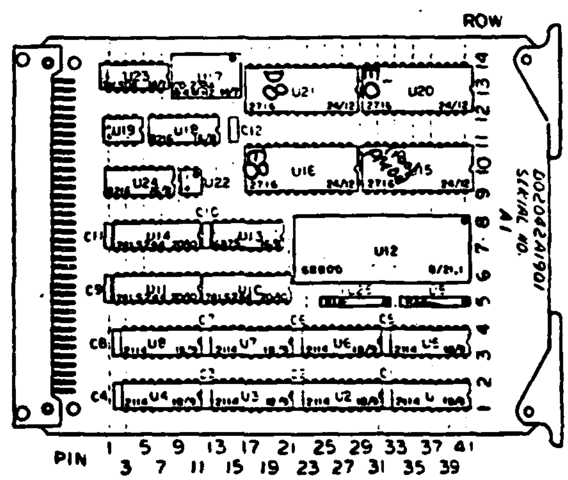


D
C
B
A
R/W-B FROM SH1
FROM SH1
D0
D1
D2
D3
DIEN
CS
U18 8216
+5VDC
GND
C35
C34
C3
C2
A1
C1
A36
C36
U19C4 6.8
U19C3 0.047
U19C2 6.8
U19C1 0.047
C1
C2
C3
C4
C5
C6
C7
C1 THRU C12 ARE 0.01UF
6
5
4





UF

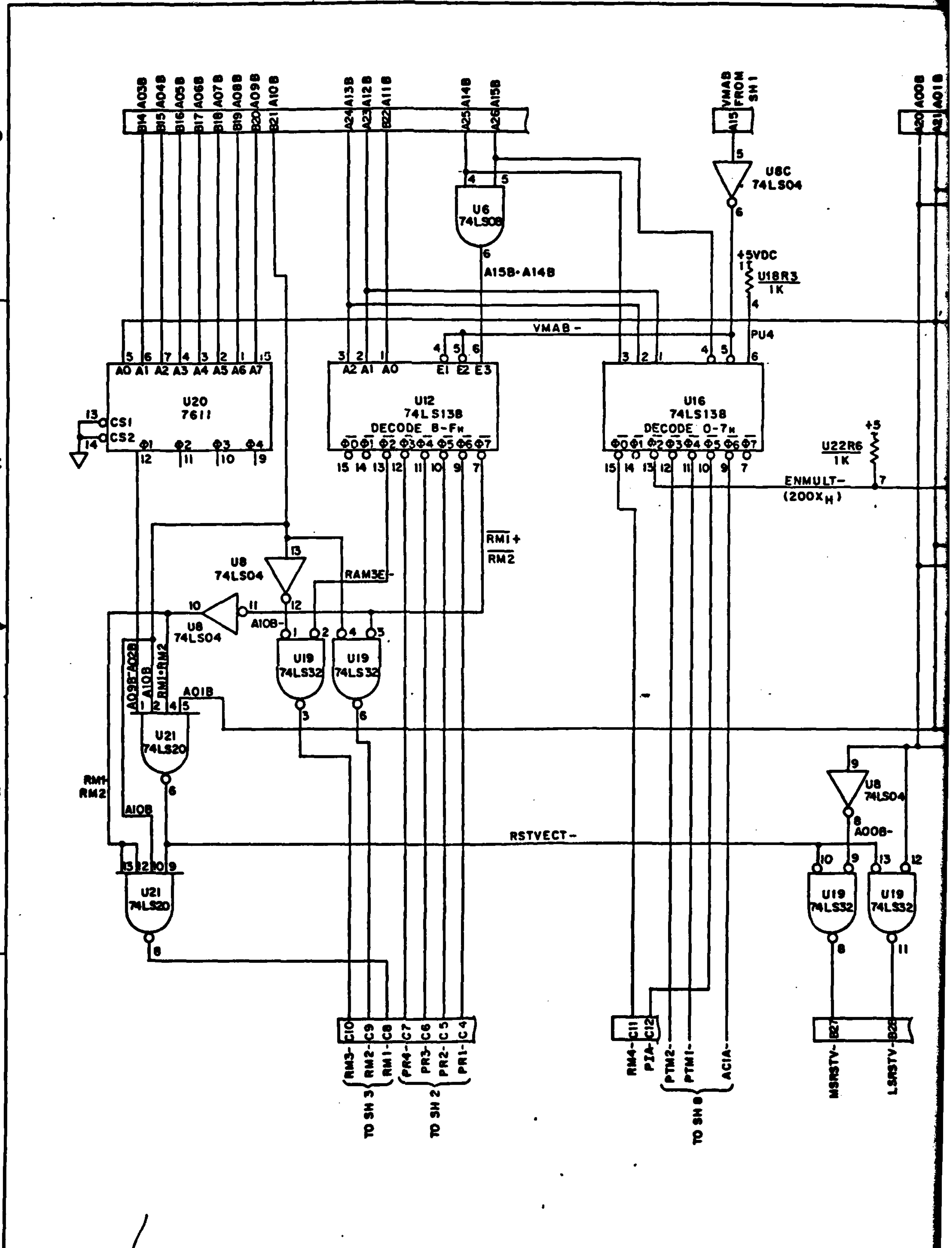


A1 MOD MPU

SIZE	CODE IDENT NO.	DWG NO.
D		

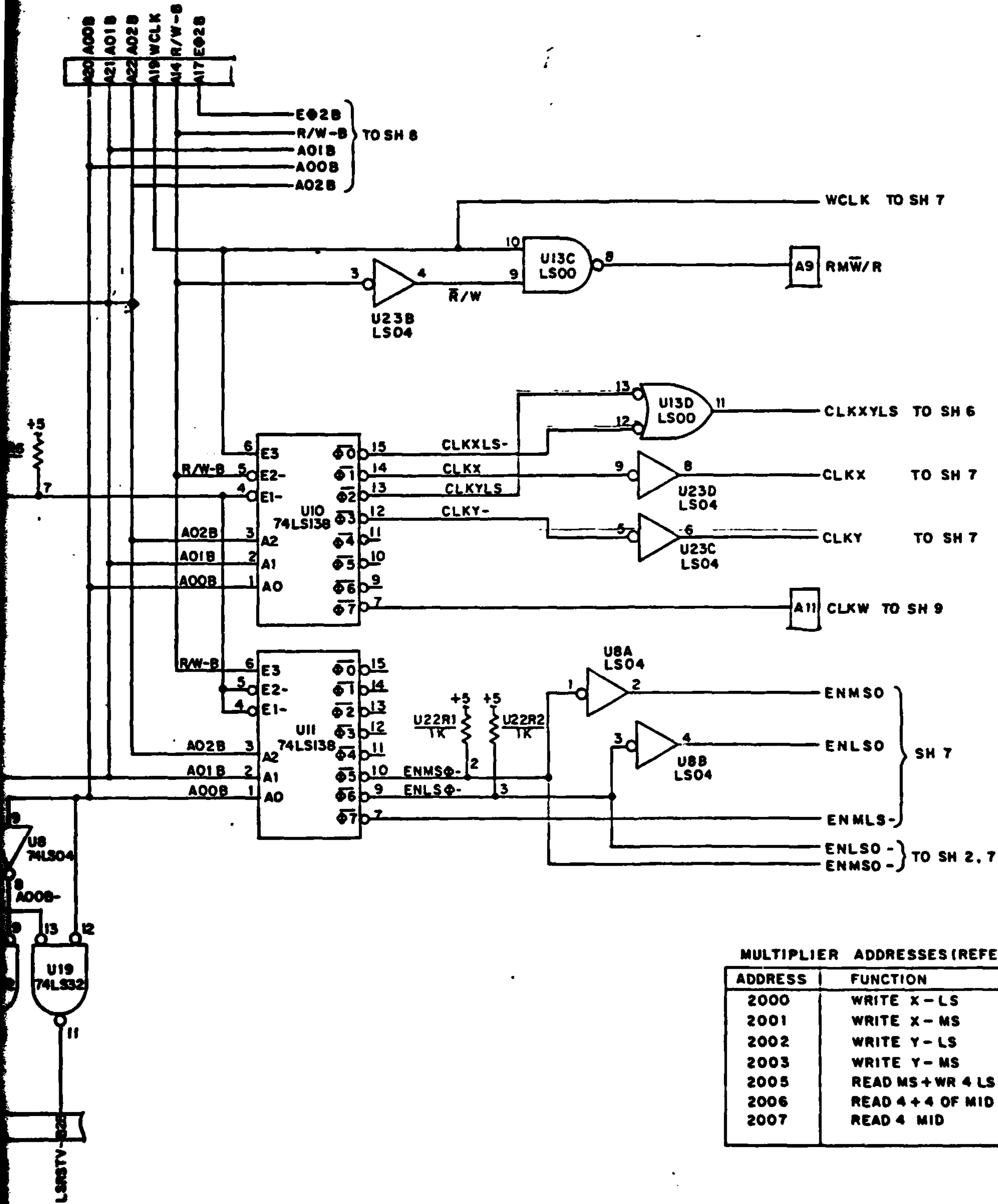
SCALE — REV SHEET

3 2 2 1



U6
74LS08
A15B·A14B
U8C
74LS04
VMAB FROM SH 1
+5VDC
U18R3
1K
VMAB-
PU4
U20
7611
U12
74LS138
DECODE 8-F H
U16
74LS138
DECODE 0-7 H
+5
U22R6
1K
ENMULT-
(200X H)
RM1+
RM2
RAM3E-
U8
74LS04
A10B-
U19
74LS32
A09B·A02B
A10B
RM1·RM2
A01B
U21
74LS20
RM1+
RM2
A10B
RSTVECT-
A00B-
RM3-
RM2-
RM1-
PR4-
PR3-
PR2-
PR1-
TO SH 3
TO SH 2
RM4-
PIA-
PTM2-
PTM1-
ACIA-
TO SH 8
MSRSTV-
LSRSTV-

6
5
4
D
C
B
A



MULTIPLIER ADDRESSES (REFERENCE ONLY)

ADDRESS	FUNCTION
2000	WRITE X-LS
2001	WRITE X-MS
2002	WRITE Y-LS
2003	WRITE Y-MS
2005	READ MS + WR 4 LS (A)
2006	READ 4 + 4 OF MID + WR 4 MID (B)
2007	READ 4 MID

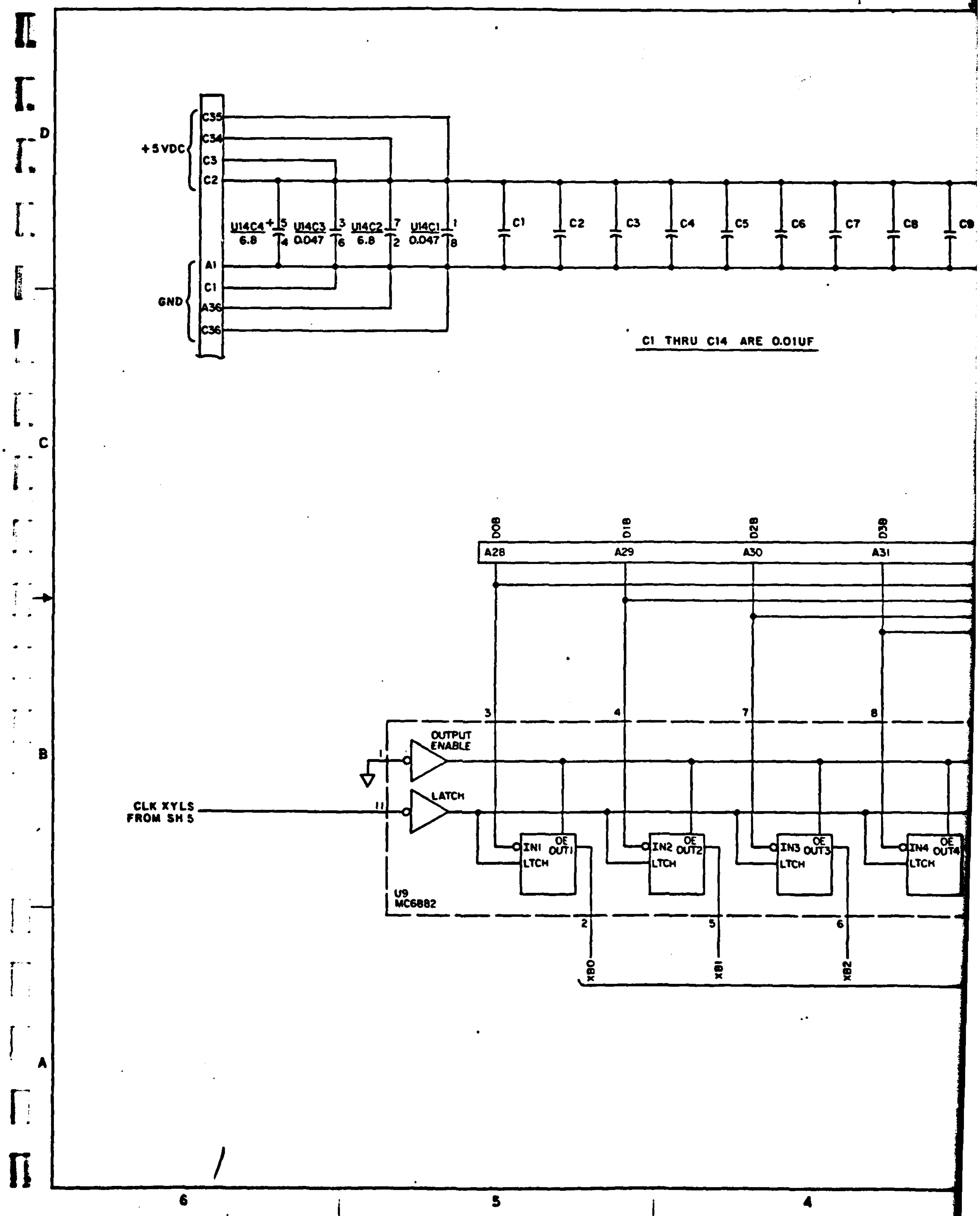
A2 MULTIPLIER ADDR DECODE & CO

SIZE D | CODE IDENT NO. | DWG NO.

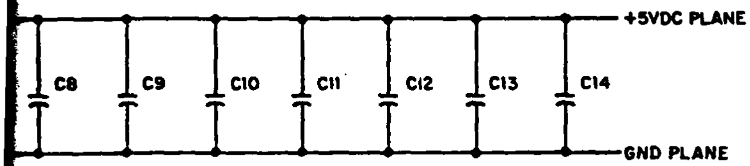
SCALE — | REV | SHEET 5

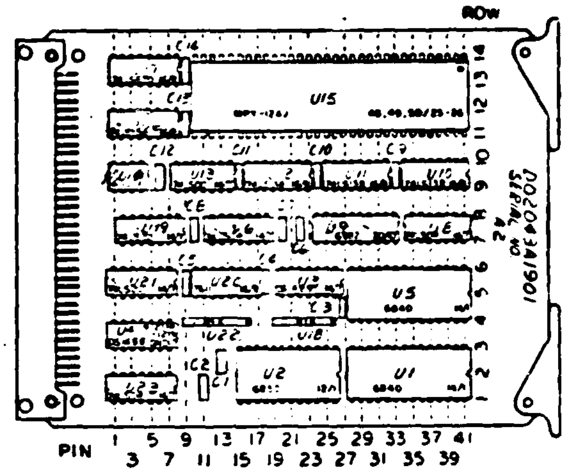
3 2 1

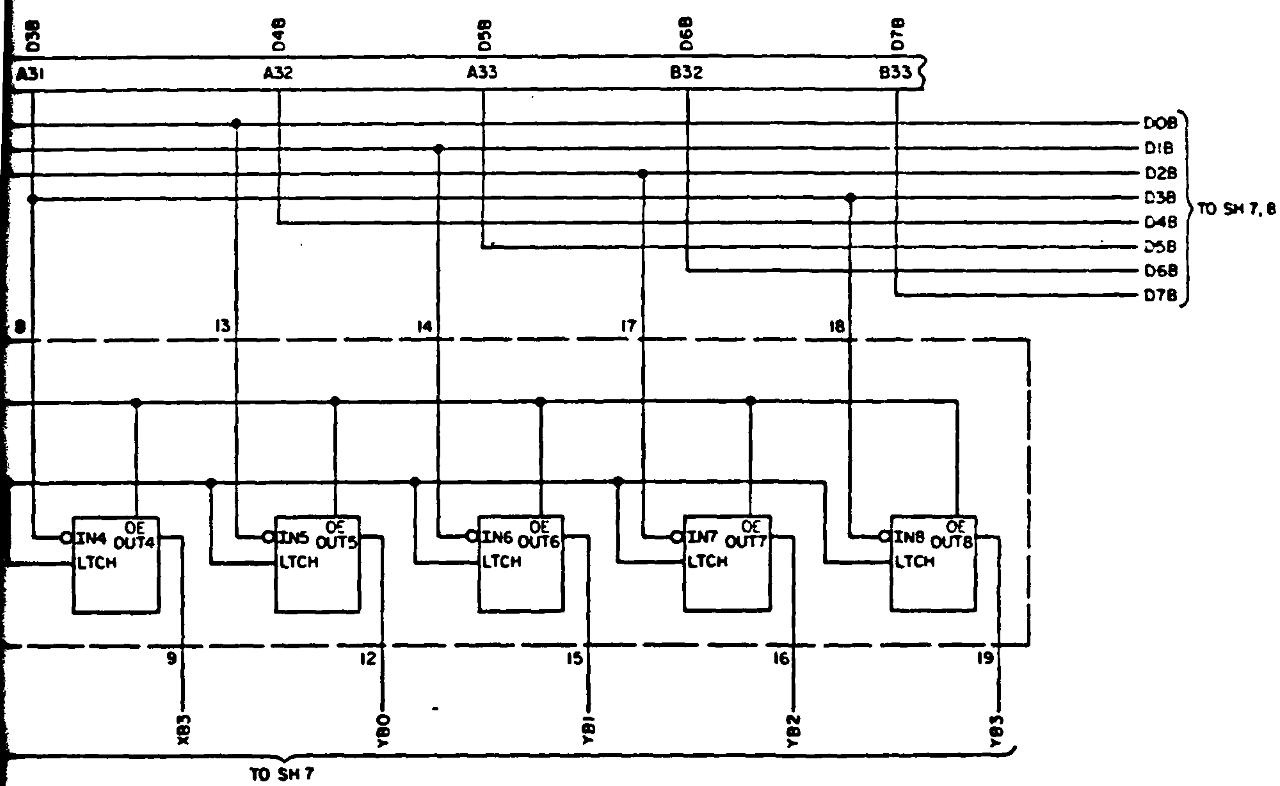
2



+5 VDC
C35
C34
C3
C2
U14C4 6.8
U14C3 0.047
U14C2 6.8
U14C1 0.047
C1
C2
C3
C4
C5
C6
C7
C8
C9
A1
C1
A36
C36
GND
C1 THRU C14 ARE 0.01UF
D0B
D1B
D2B
D3B
A28
A29
A30
A31
OUTPUT ENABLE
LATCH
CLK XYLS FROM SH 5
IN1
OUT1
OE
LTCH
IN2
OUT2
IN3
OUT3
IN4
OUT4
U9
MC6882
XB0
XB1
XB2
D
C
B
A
6
5
4



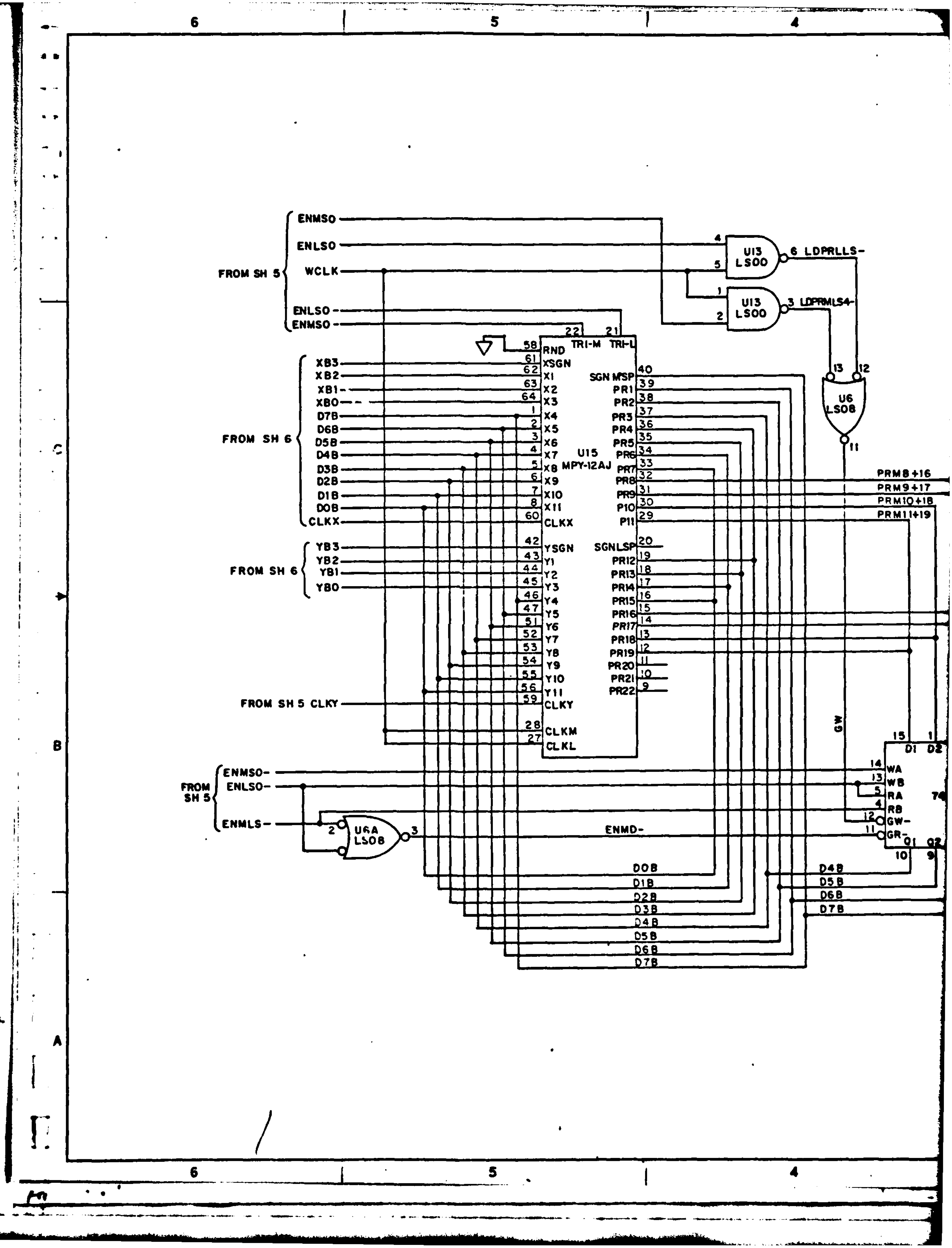




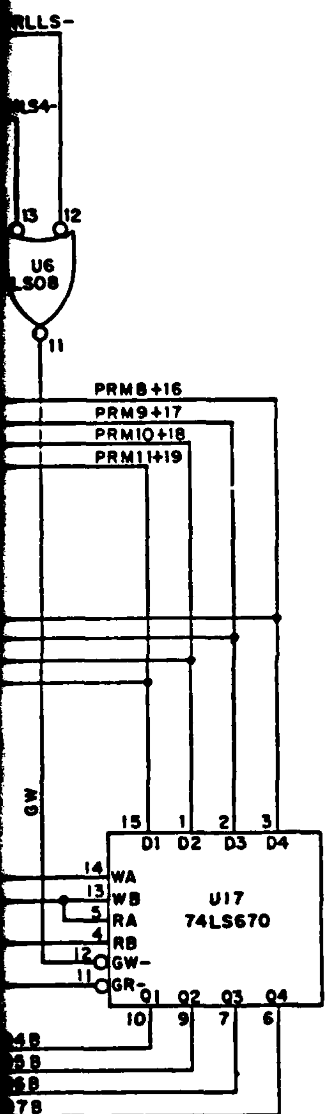
A2 MULTIPLIER

SIZE	CODE IDENT NO	DWG NO
D		
SCALE —	REV	SHEET 6

3 2 1



6
5
4
ENMSO
ENLSO
FROM SH 5
WCLK
ENLSO
ENMSO
U13
LS00
6 LDPRLLS-
U13
LS00
3 LDPRMLS4-
U6
LS08
TRI-M
TRI-L
RND
XSGN
FROM SH 6
XB3
XB2
XB1-
XB0
D7B
D6B
D5B
D4B
D3B
D2B
D1B
D0B
CLKX
U15
MPY-12AJ
SGN MSP
SGNLSP
YB3
YB2
YB1
YB0
YSGN
CLKY
FROM SH 5 CLKY
CLKM
CLKL
PRM8+16
PRM9+17
PRM10+18
PRM11+19
GW
FROM SH 5
ENMSO-
ENLSO-
ENMLS-
U6A
LS08
ENMD-
WA
WB
RA
RB
GW-
GR-
D0B
D1B
D2B
D3B
D4B
D5B
D6B
D7B
C
B
A

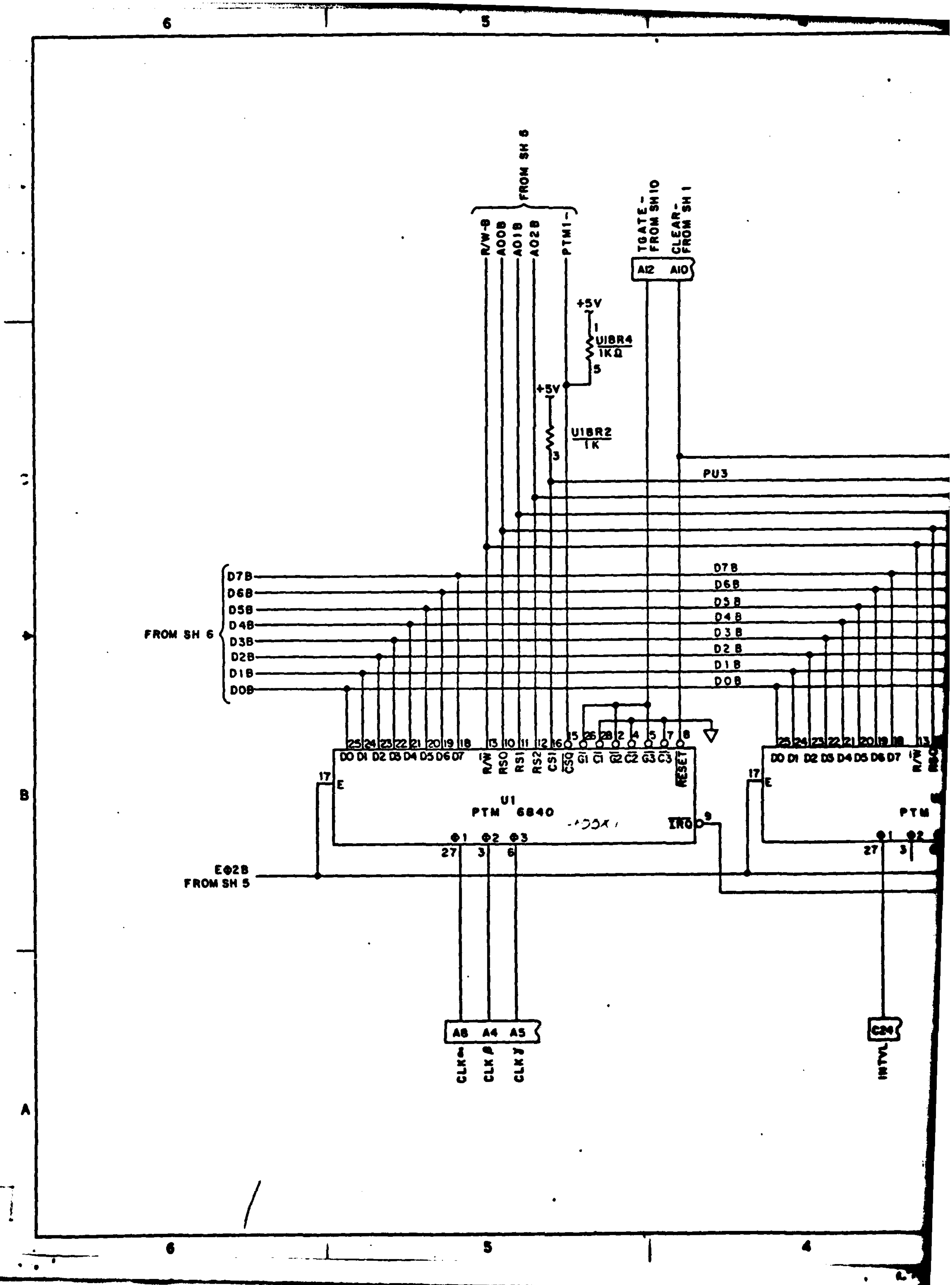


A2 MULTIPLIER

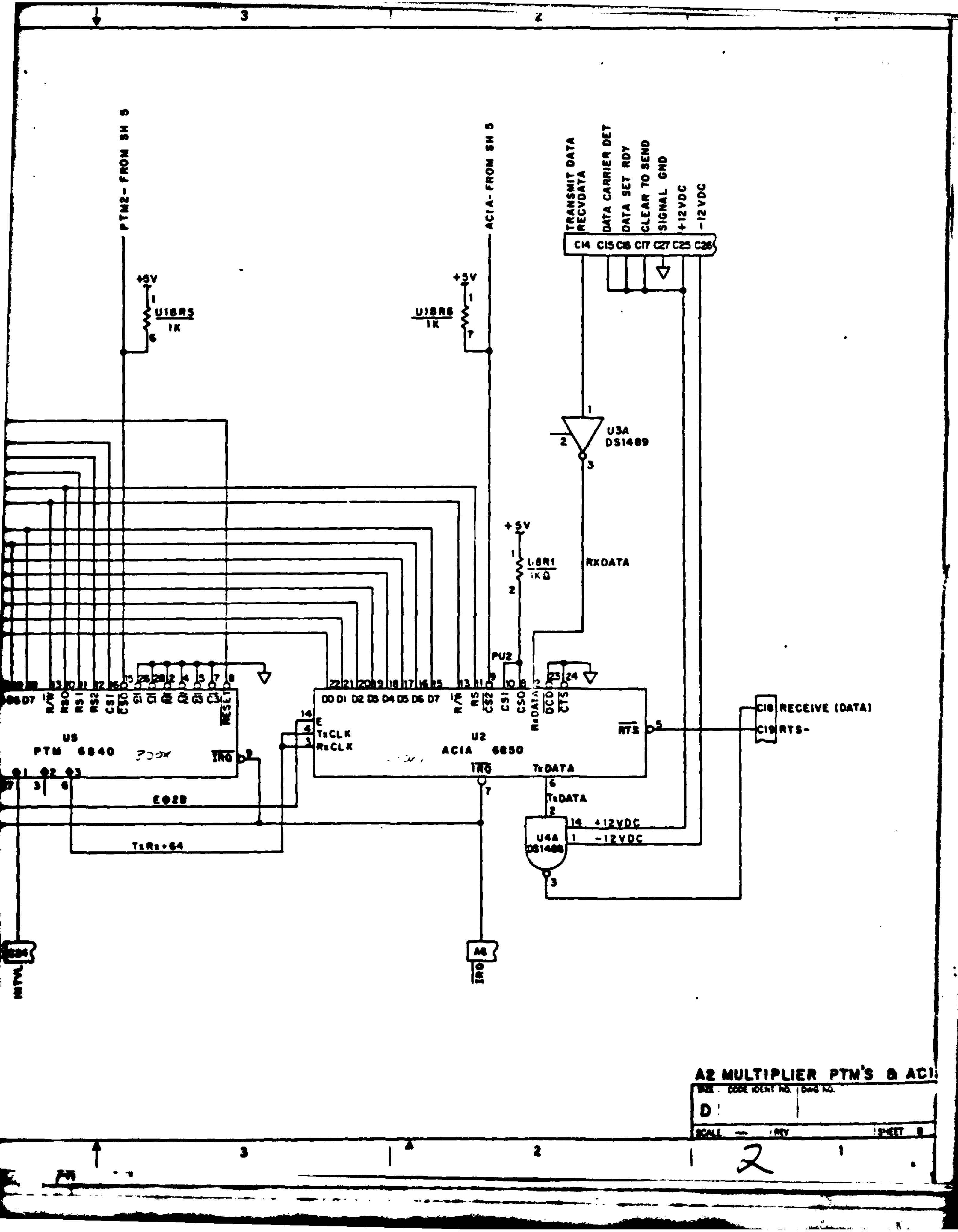
SIZE	CODE IDENT NO.	DWG NO.
D		

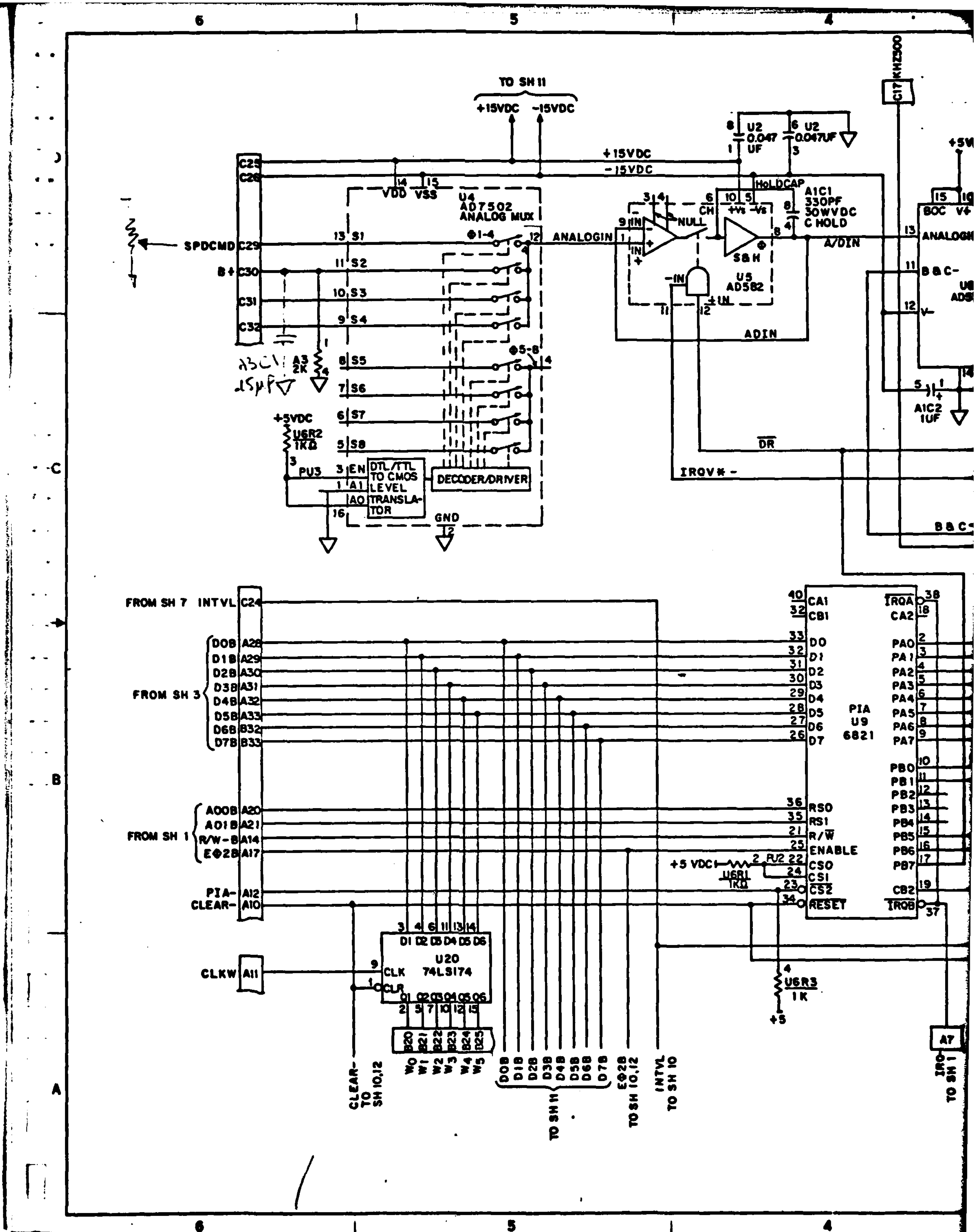
SCALE — REV SHEET 7

2

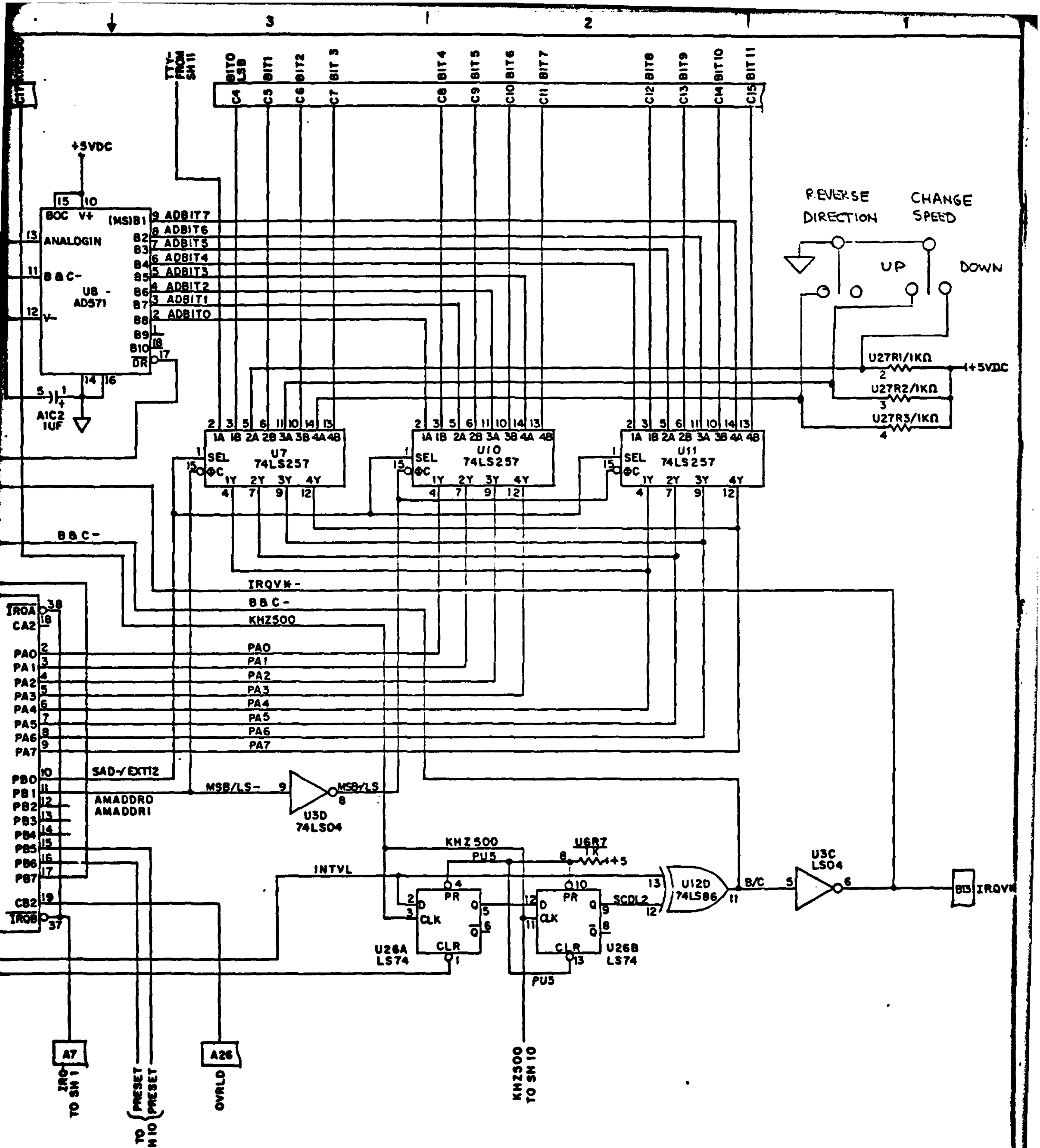


FROM SH 6
R/W-B
A00B
A01B
A02B
PTM1-
TGATE-FROM SH10
CLEAR-FROM SH 1
A12
A10
+5V
U18R4
1KΩ
U18R2
1K
PU3
D7B
D6B
D5B
D4B
D3B
D2B
D1B
D0B
FROM SH 6
U1
PTM 6840
E⊕2B
FROM SH 5
A6
A4
A5
CLK
INTVL
C24
PTM


PTM2-FROM SH 5
ACIA-FROM SH 5
TRANSMIT DATA
RECVDATA
DATA CARRIER DET
DATA SET RDY
CLEAR TO SEND
SIGNAL GND
+12VDC
-12VDC
C14 C15 C16 C17 C27 C25 C26
+5V
U18R5
1K
U18R6
U3A
DS1489
RXDATA
U8R1
1KΩ
PU2
U5
PTM 6840
U2
ACIA 6850
TxCLK
RxCLK
RTS
TxDATA
IRQ
RESET
C18 RECEIVE (DATA)
C19 RTS-
U4A
DS1488
+12VDC
-12VDC
E02B
TxRx÷64
A6
IRQ
INTVL
A2 MULTIPLIER PTM'S & ACI
SIZE
CODE IDENT NO.
DWG NO.
D
SCALE
REV
SHEET
3
2
1



TO SH 11
+15VDC
-15VDC
C25
C26
+15VDC
-15VDC
U2 0.047 UF
U2 0.047UF
HOLDCAP
A1C1 330PF 30WVDC C HOLD
+5V
C17 KHZ500
VDD VSS
U4 AD7502 ANALOG MUX
SPDCMD C29
B+ C30
C31
C32
S1
S2
S3
S4
S5
S6
S7
S8
Φ1-4
Φ5-8
ANALOGIN
NULL
CH
+Vs
-Vs
S&H
U5 AD582
-IN
+IN
A/DIN
ADIN
BOC V+
ANALOGIN
B&C-
V-
A1C2 1UF
A3 2K
+5VDC
U6R2 1KΩ
PU3
EN
A1
A0
DTL/TTL TO CMOS LEVEL TRANSLATOR
DECODER/DRIVER
GND
DR
IRQV* -
B&C-
FROM SH 7 INTVL C24
FROM SH 3
D0B A28
D1B A29
D2B A30
D3B A31
D4B A32
D5B A33
D6B B32
D7B B33
FROM SH 1
A00B A20
A01B A21
R/W-B A14
EΦ2B A17
PIA-CLEAR- A12 A10
CLKW A11
CA1
CB1
D0 D1 D2 D3 D4 D5 D6 D7
RS0
RS1
R/W
ENABLE
CS0
CS1
CS2
RESET
PIA U9 6821
IRQA
CA2
PA0 PA1 PA2 PA3 PA4 PA5 PA6 PA7
PB0 PB1 PB2 PB3 PB4 PB5 PB6 PB7
CB2
IRQB
+5 VDC
U6R1 1KΩ
PU2
U6R3 1K
+5
U20 74LS174
CLK
CLR
D1 D2 D3 D4 D5 D6
Q1 Q2 Q3 Q4 Q5 Q6
B20 B21 B22 B23 B24 B25
W0 W1 W2 W3 W4 W5
CLEAR- TO SH 10,12
D0B D1B D2B D3B D4B D5B D6B D7B
TO SH 11
EΦ2B TO SH 10,12
INTVL TO SH 10
A7
IRQ TO SH 1

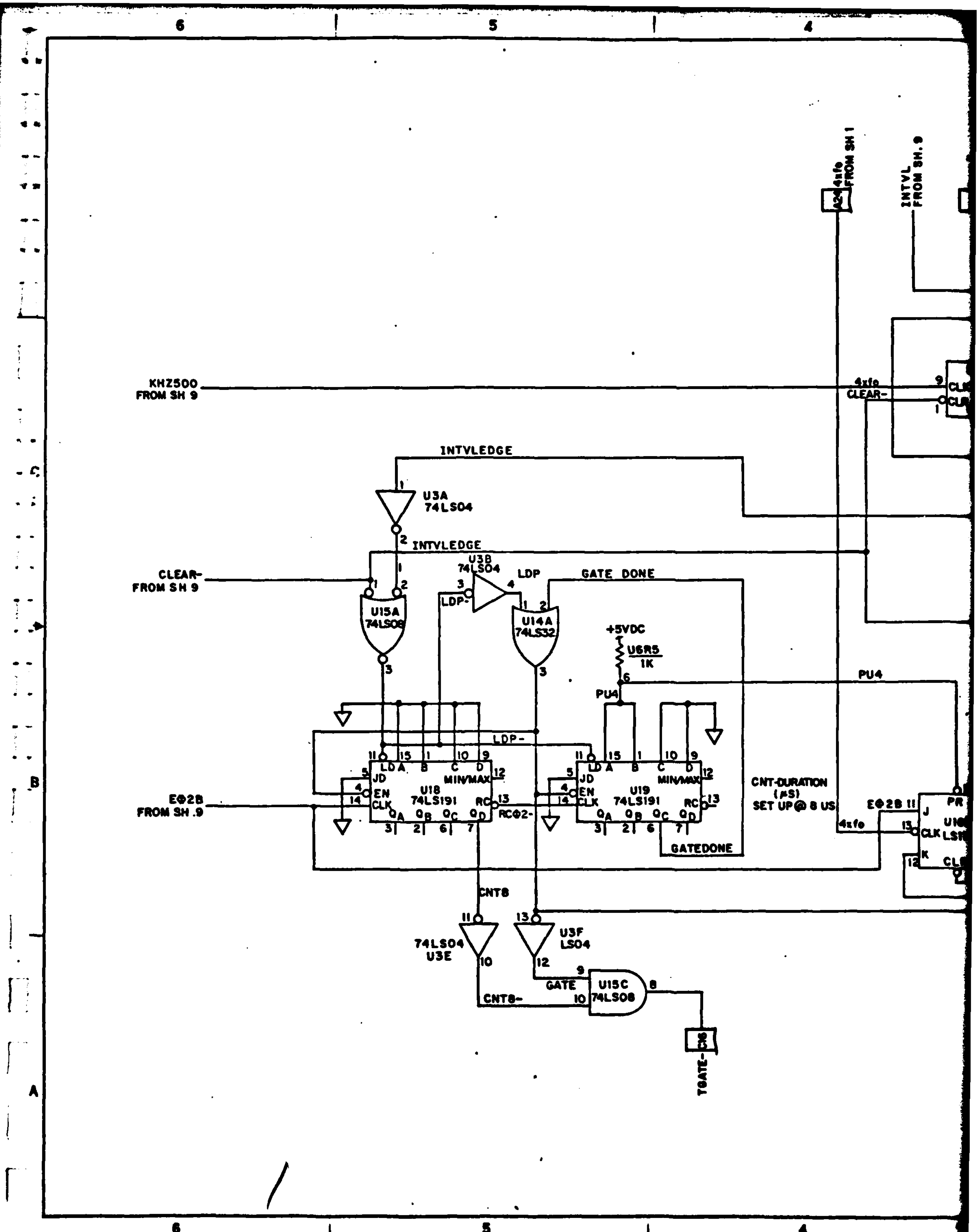


A3 PIA; DATA ACQUISITION

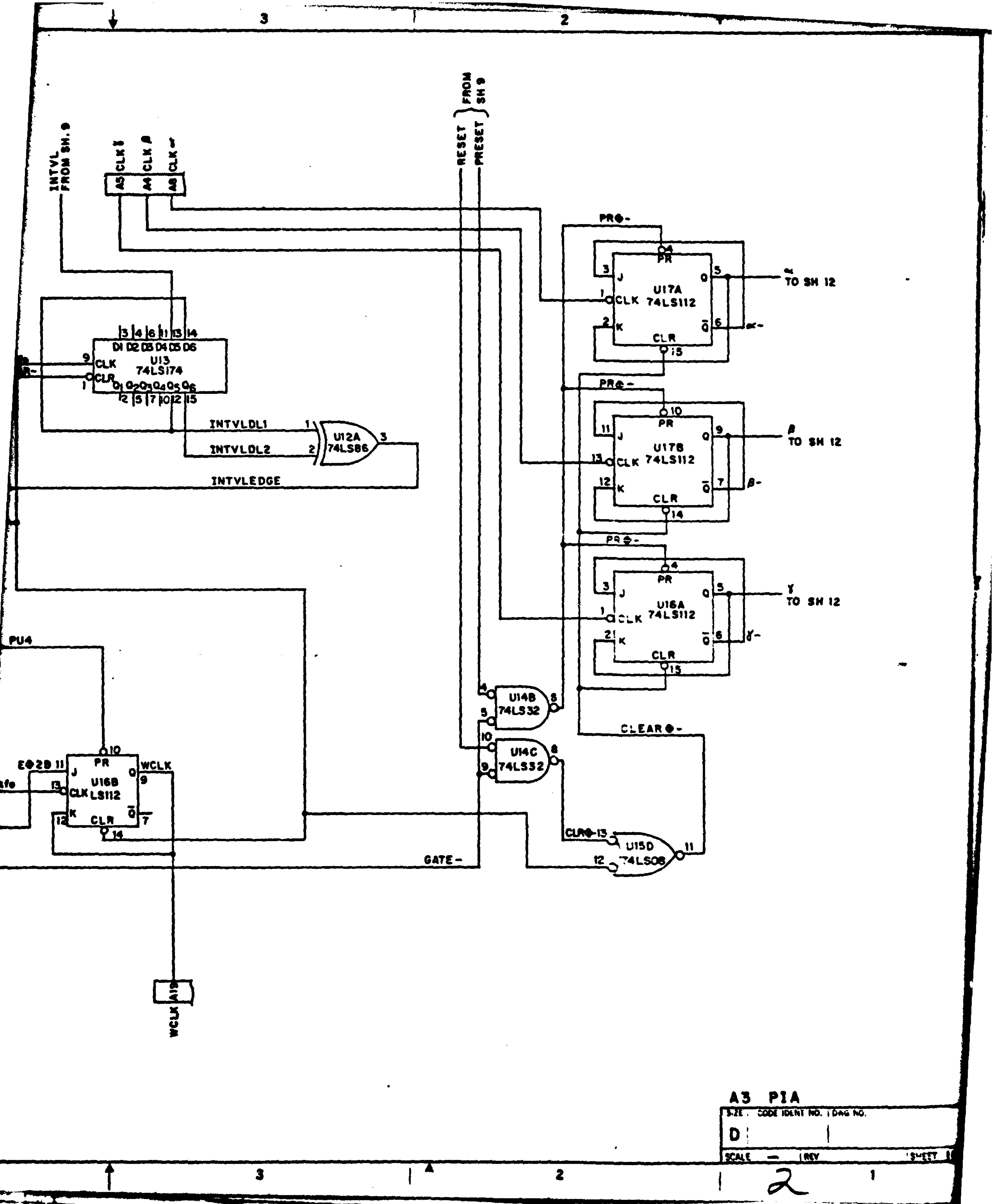
SIZE D | CODE IDENT NO. | DWG NO.

SCALE — | REV | SHEET 9

2



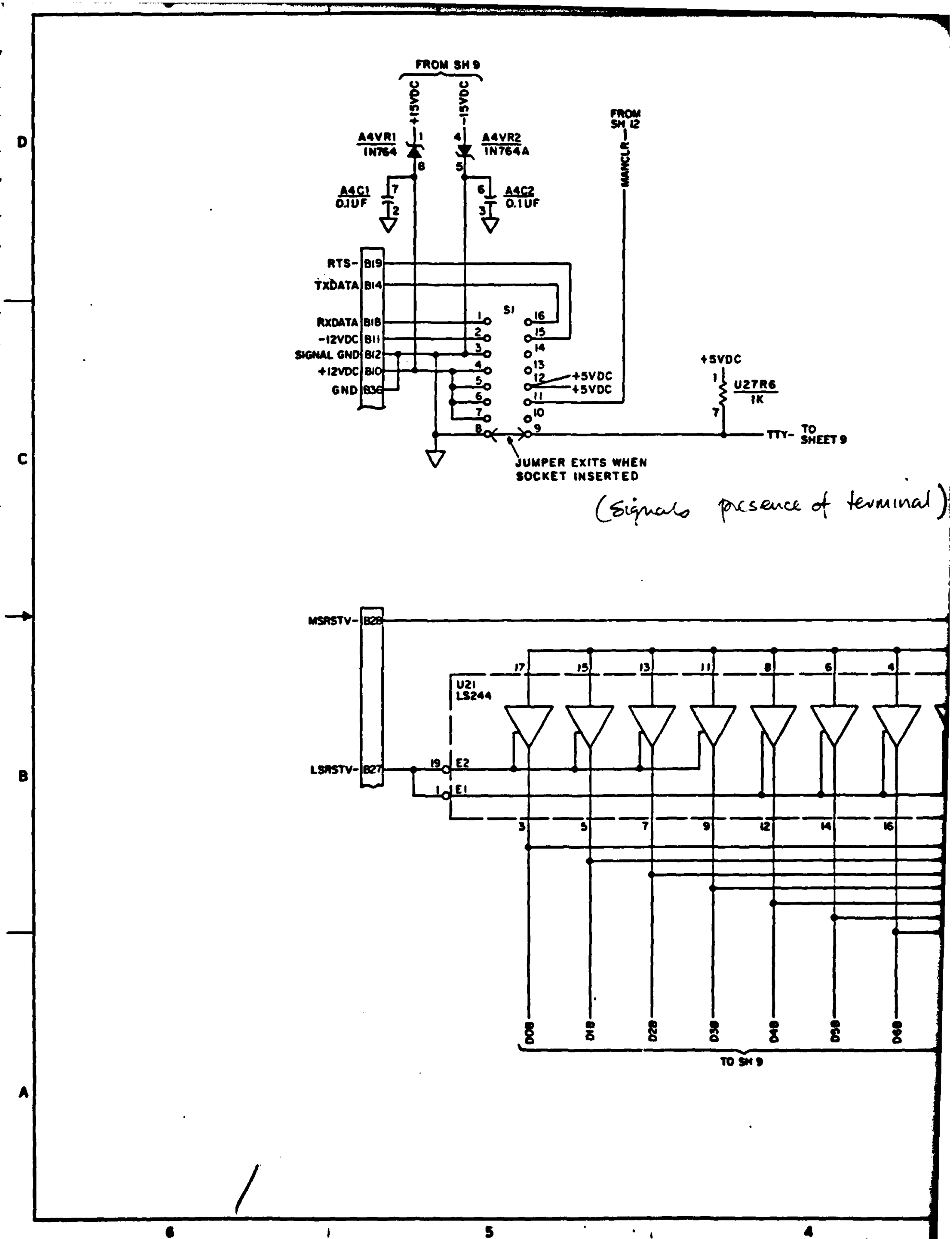
KHZ500
FROM SH 9
4xfo
FROM SH 1
INTVL
FROM SH. 9
4xfo
CLEAR-
INTVLEDGE
U3A
74LS04
INTVLEDGE
CLEAR-
FROM SH 9
U3B
74LS04
LDP
LDP-
GATE DONE
U15A
74LS08
U14A
74LS32
+5VDC
U6R5
1K
PU4
PU4
LDP-
U18
74LS191
MIN/MAX
U19
74LS191
MIN/MAX
EΦ2B
FROM SH .9
RCΦ2-
GATEDONE
CNT-DURATION
(μS)
SET UP@ 8 US
EΦ2B
4xfo
CNT8
74LS04
U3E
U3F
LS04
GATE
U15C
74LS08
CNT8-
TGATE-



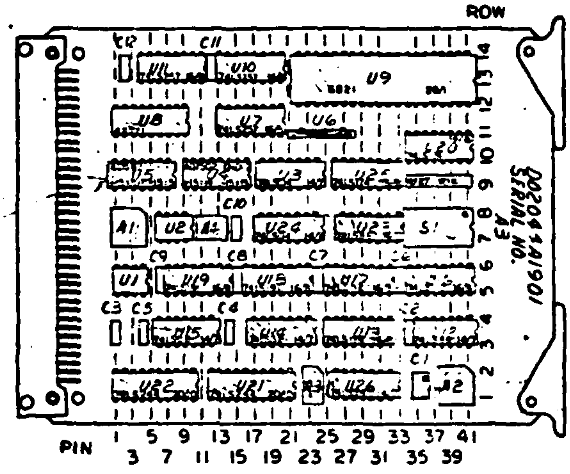
A3 PIA

SIZE	CODE IDENT NO.	DWG NO.
D		
SCALE —	REV	SHEET

2

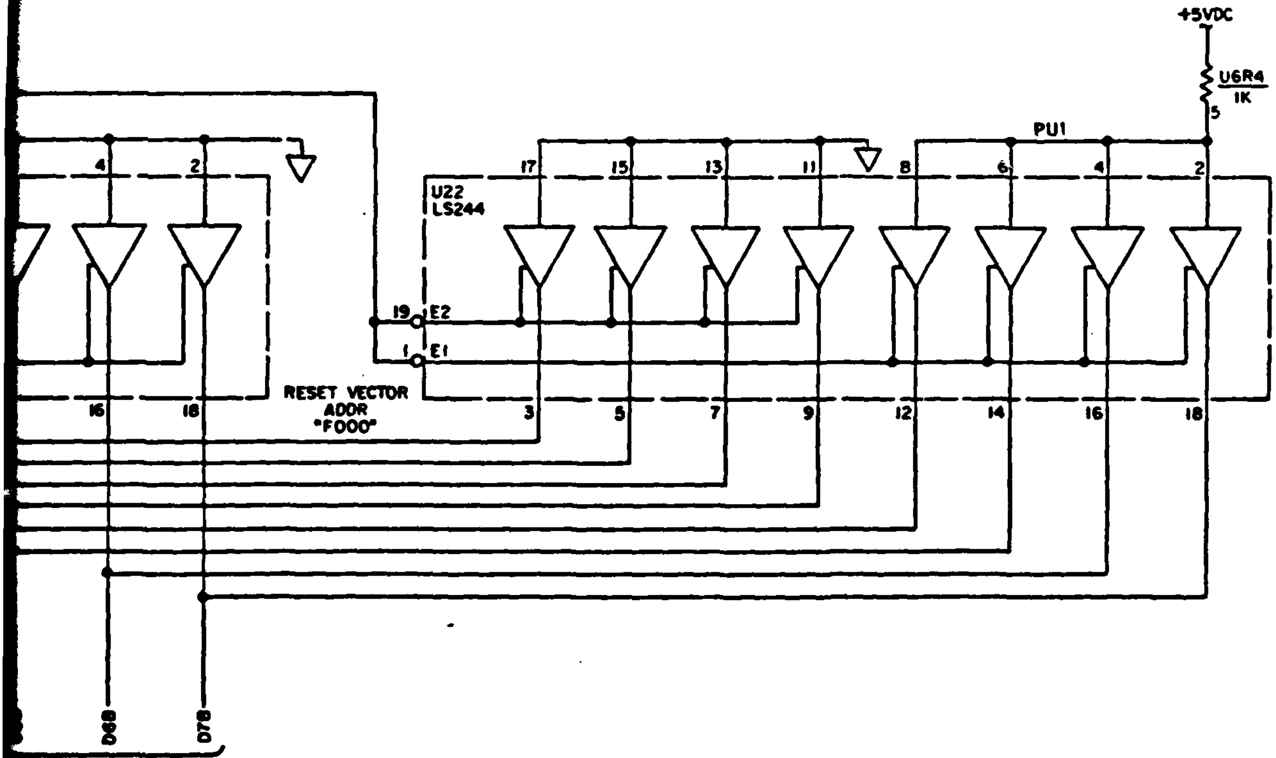

FROM SH 9
+15VDC
-15VDC
A4VR1
1N764
A4VR2
1N764A
A4C1
0.1UF
A4C2
0.1UF
FROM SH 12
MANCLR
RTS-
TXDATA
RXDATA
-12VDC
SIGNAL GND
+12VDC
GND
B19
B14
B18
B11
B12
B10
B36
S1
+5VDC
U27R6
1K
TTY-
TO SHEET 9
JUMPER EXITS WHEN SOCKET INSERTED
(Signals presence of terminal)
MSRSTV-
B28
LSRSTV-
B27
U21
LS244
E2
E1
D0B
D1B
D2B
D3B
D4B
D5B
D6B
TO SH 9
D
C
B
A
6
5
4

terminal)



15

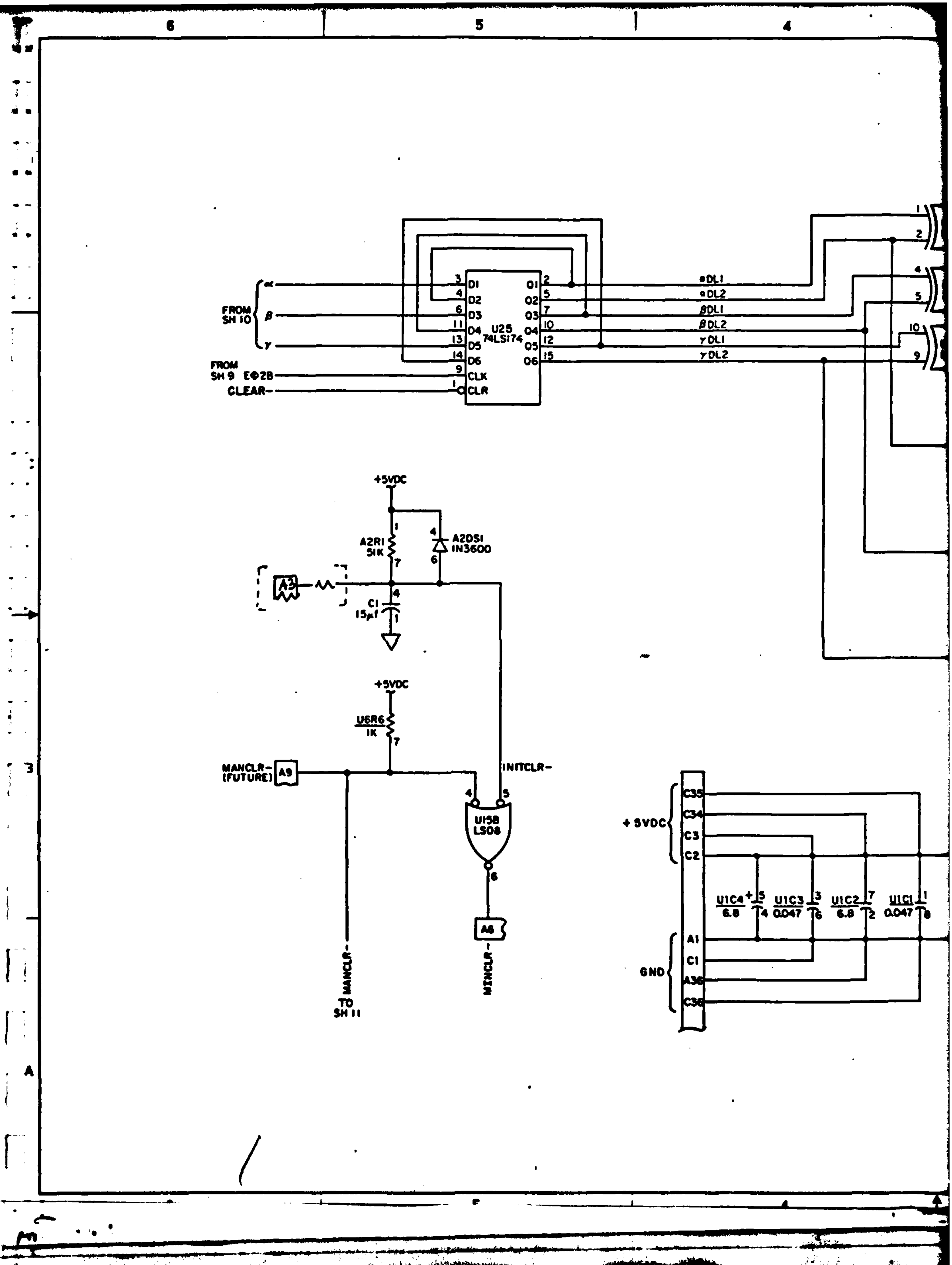
U1 & U2 = SPRAGUE 939Z685X0035DE



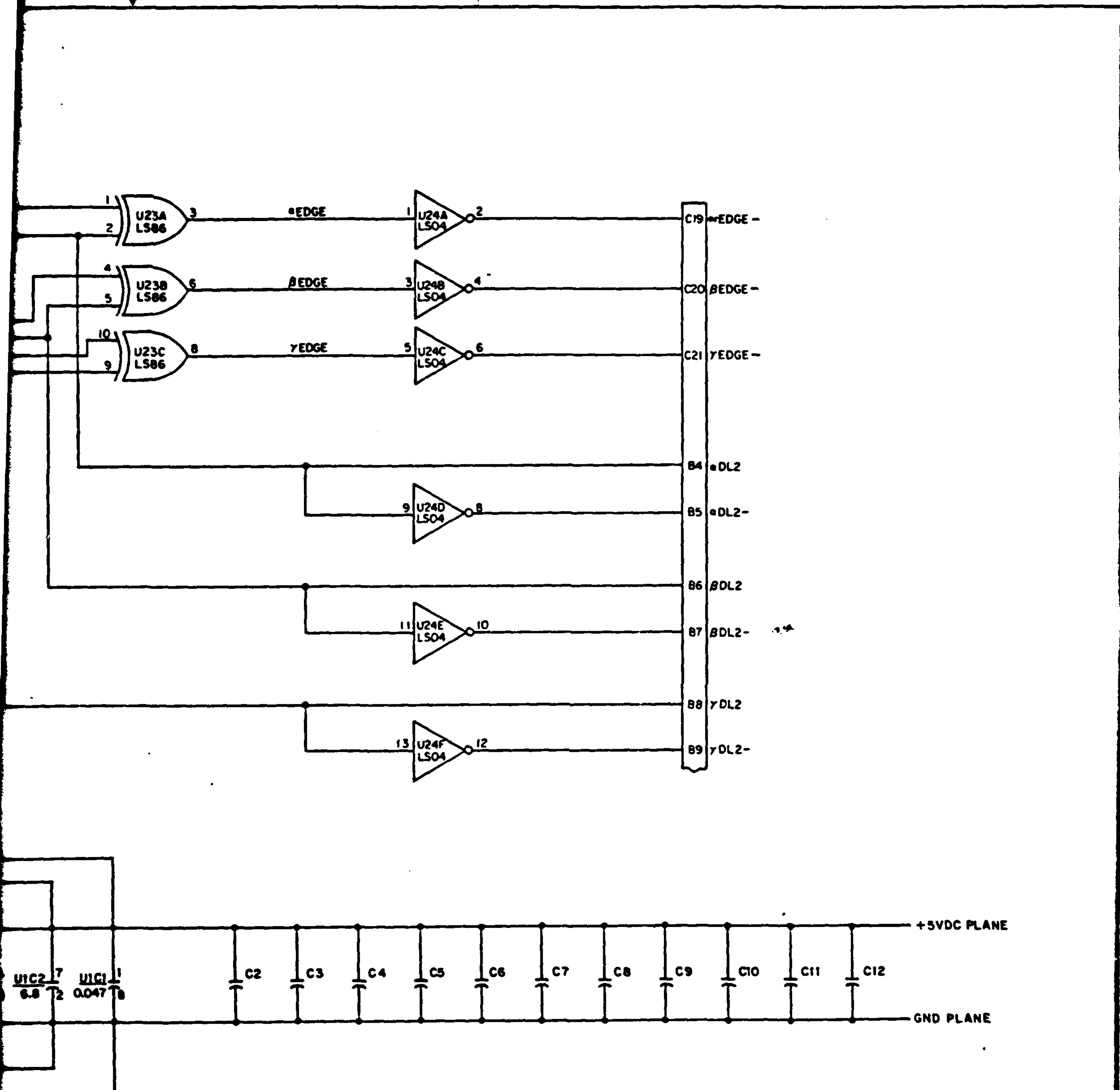
A3 PIA

SIZE	CODE IDENT NO.	DWG NO.
D		
SCALE —	REV	SHEET

2



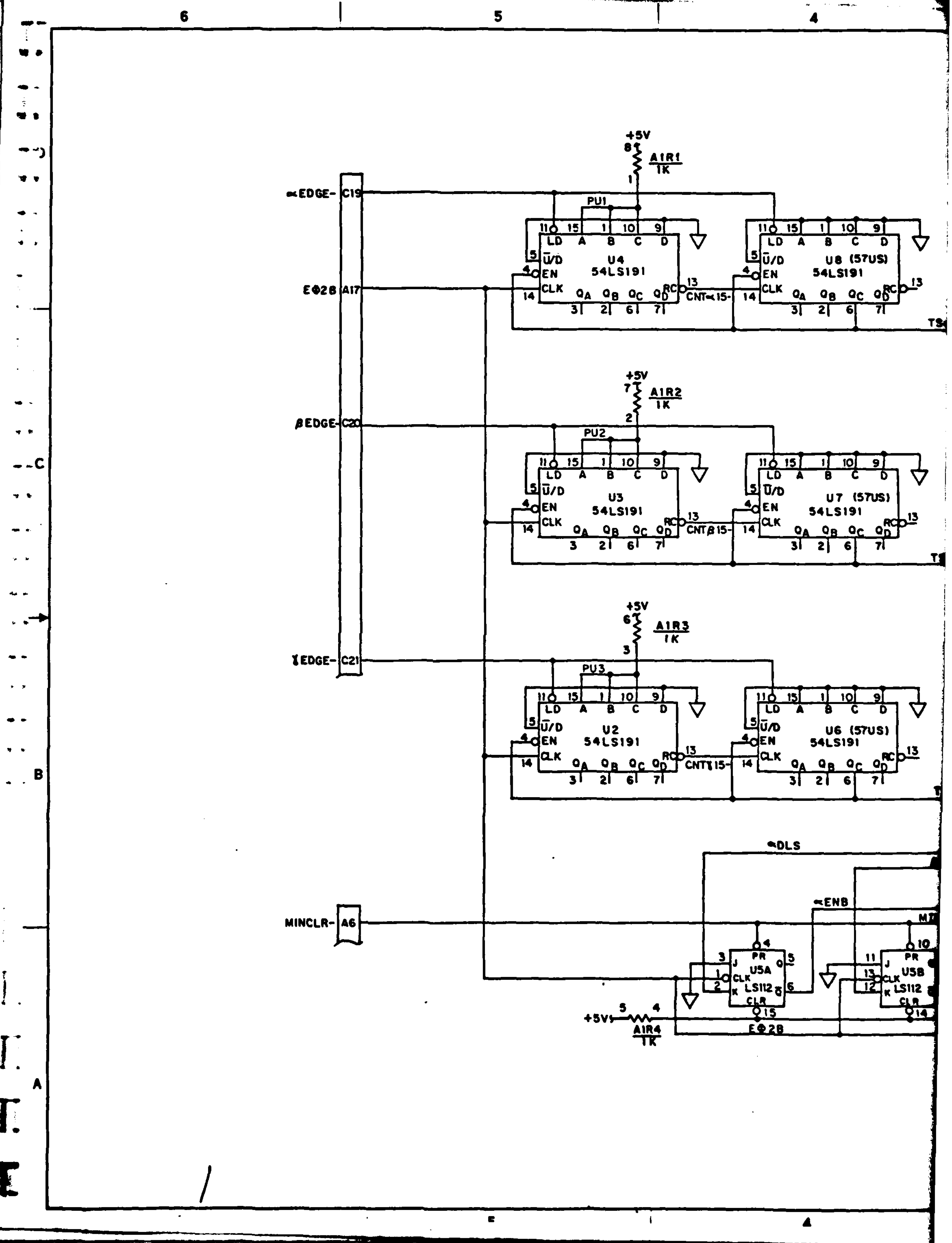
6
5
4
FROM SH 10
α
β
γ
FROM SH 9 EΦ2B
CLEAR-
D1
D2
D3
D4
D5
D6
CLK
CLR
Q1
Q2
Q3
Q4
Q5
Q6
U25
74LS174
αDL1
αDL2
βDL1
βDL2
γDL1
γDL2
+5VDC
A2R1
51K
A2DS1
1N3600
A3
C1
15μf
+5VDC
U6R6
1K
MANCLR-
(FUTURE)
A9
INITCLR-
U15B
LS08
A6
MINCLR-
MANCLR-
TO SH 11
+5VDC
C35
C34
C3
C2
U1C4
6.8
U1C3
0.047
U1C2
6.8
U1C1
0.047
GND
A1
C1
A36
C36
3
A



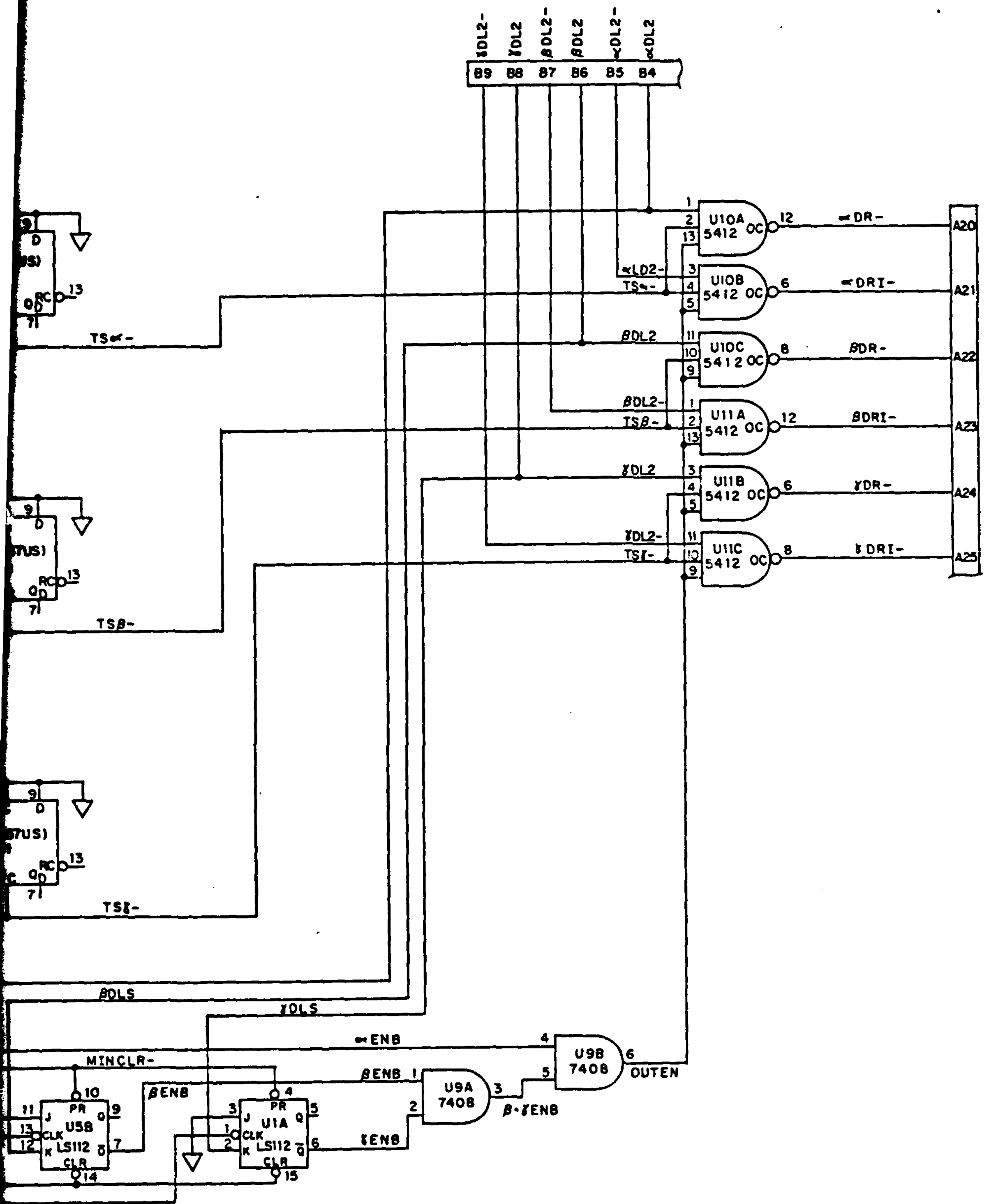
C2 THRU C12 ARE 0.01UF

A3 PIA - S.T. LOGIC(PART OF)A3

SIZE D	CODE IDENT NO.	DWG NO.
SCALE —	REV	SHEET 12



6
5
4
+5V
A1R1
1K
αEDGE- C19
PU1
U4
54LS191
U8 (57US)
54LS191
EΦ2B A17
CNTα15-
+5V
A1R2
1K
βEDGE- C20
PU2
U3
54LS191
U7 (57US)
54LS191
CNTβ15-
+5V
A1R3
1K
γEDGE- C21
PU3
U2
54LS191
U6 (57US)
54LS191
CNTγ15-
αDLS
αENB
MINCLR- A6
U5A
LS112
U5B
LS112
+5V
A1R4
1K
EΦ2B
C
B
A

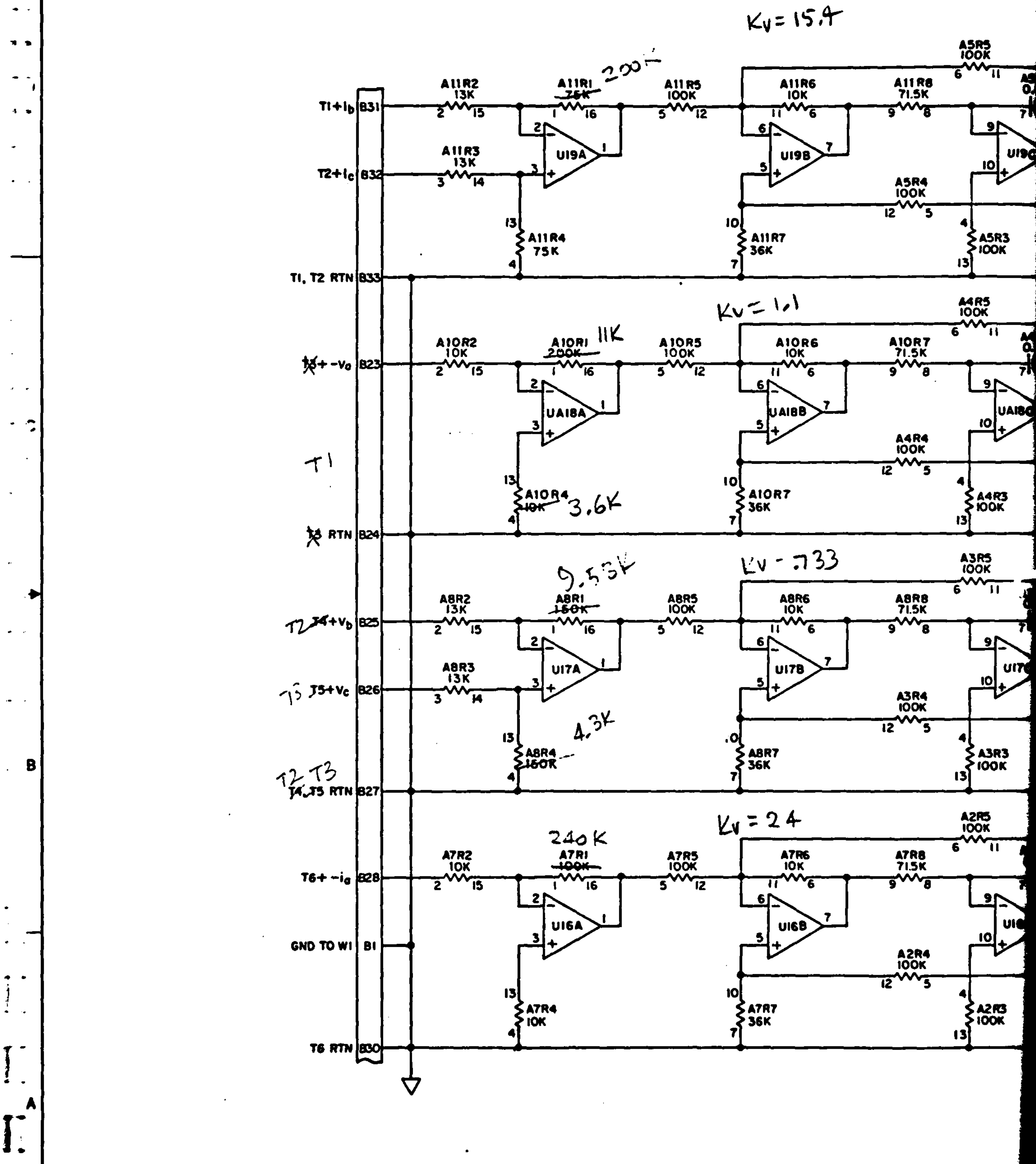


A4 S.T. & FILTERS

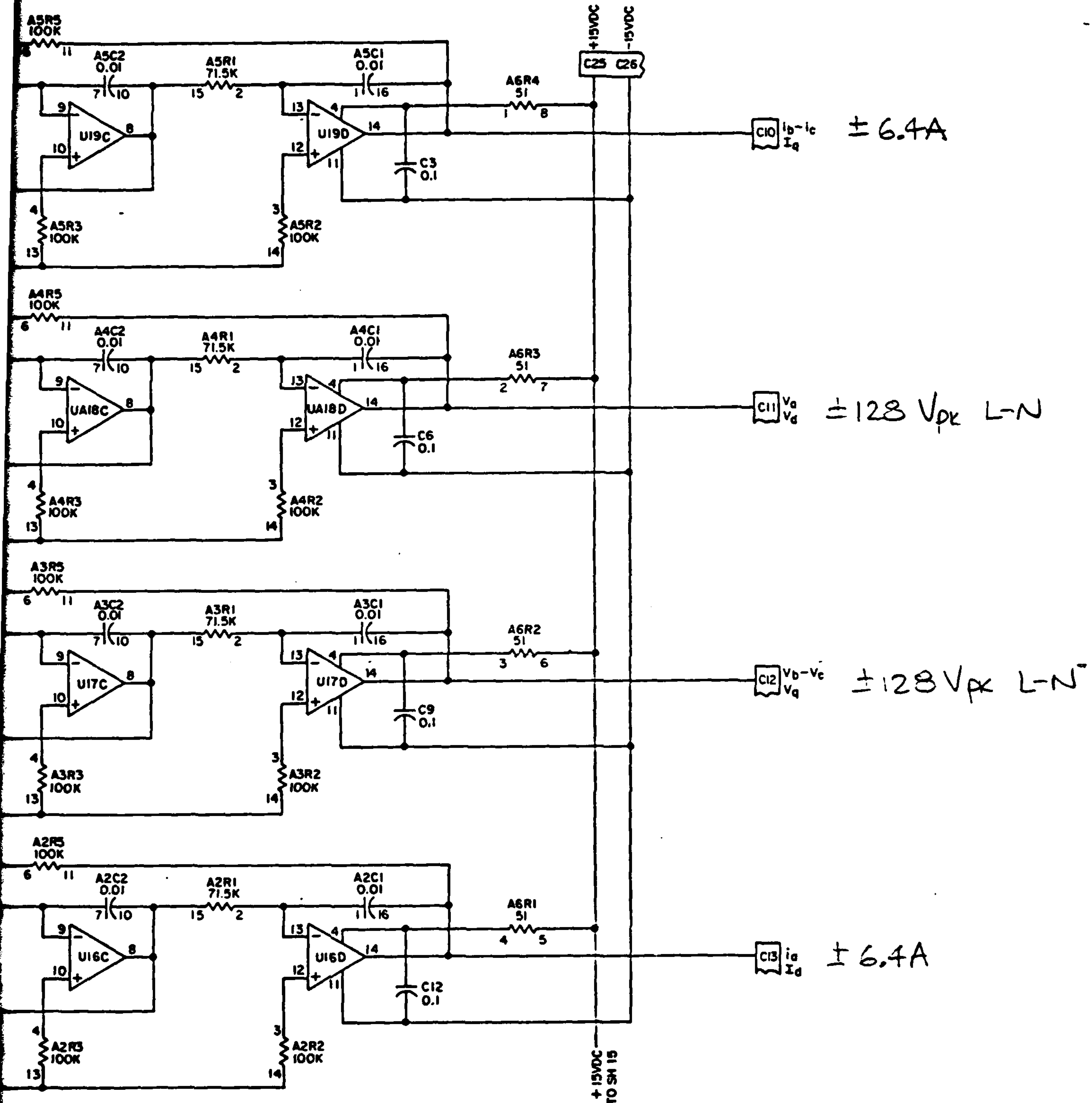
SIZE	CODE IDENT NO	DWG NO.
D		

SCALE — REV SHEET

2



NOTE: A7R1, A8R1, A10R1, A11R1 ESTABLISH FEEDBACK SCALING.



A4 S.T. & LPF

SIZE	CODE IDENT NO.	DWG NO.
D		
SCALE —	REV	SHEET 14

16. (APPLICATION DEPENDENT)

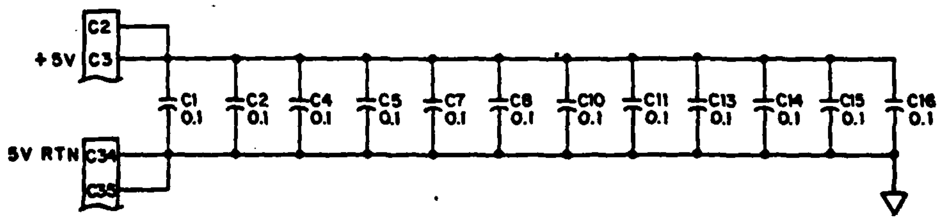
3 2 1

2

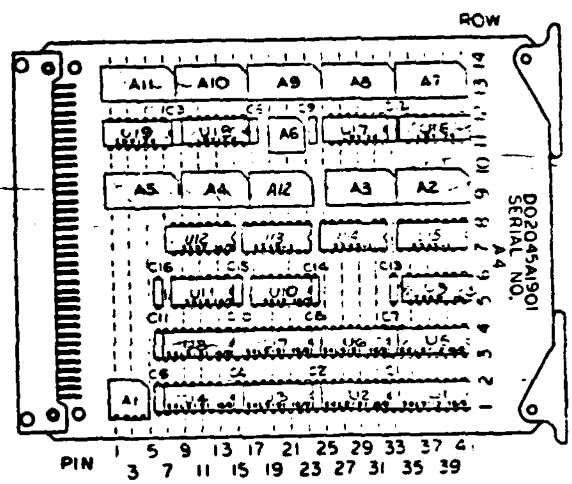
6 5 4

B

A


C2
+5V C3
C1 0.1
C2 0.1
C4 0.1
C5 0.1
C7 0.1
C8 0.1
C10 0.1
C11 0.1
C13 0.1
C14 0.1
C15 0.1
C16 0.1
5V RTN C34
C35

6 5 4



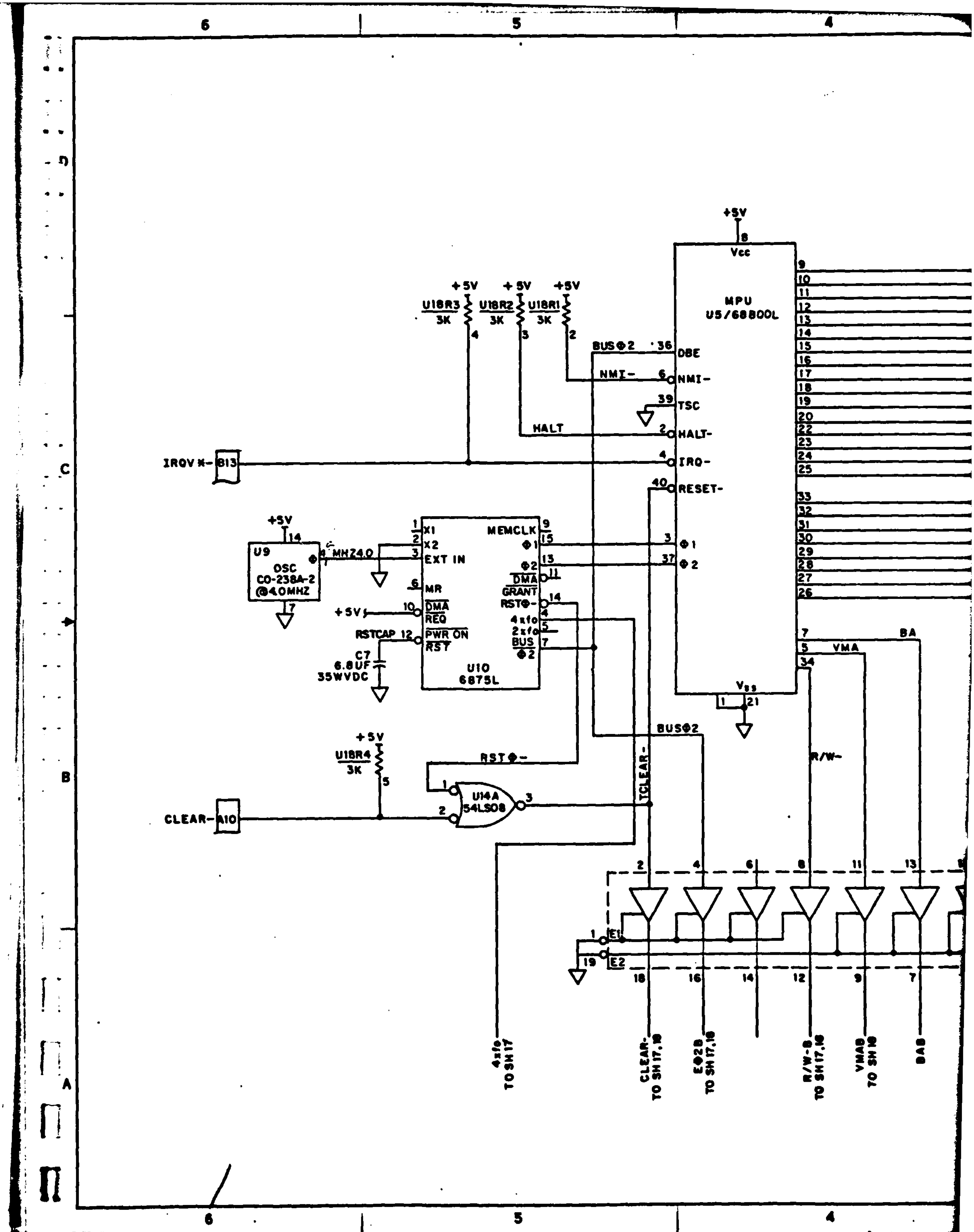
NOTE: ON LM124 PIN 4 = V+ (+15VDC)
PIN 11= V- (-15VDC)

A4 S.T. & LPF

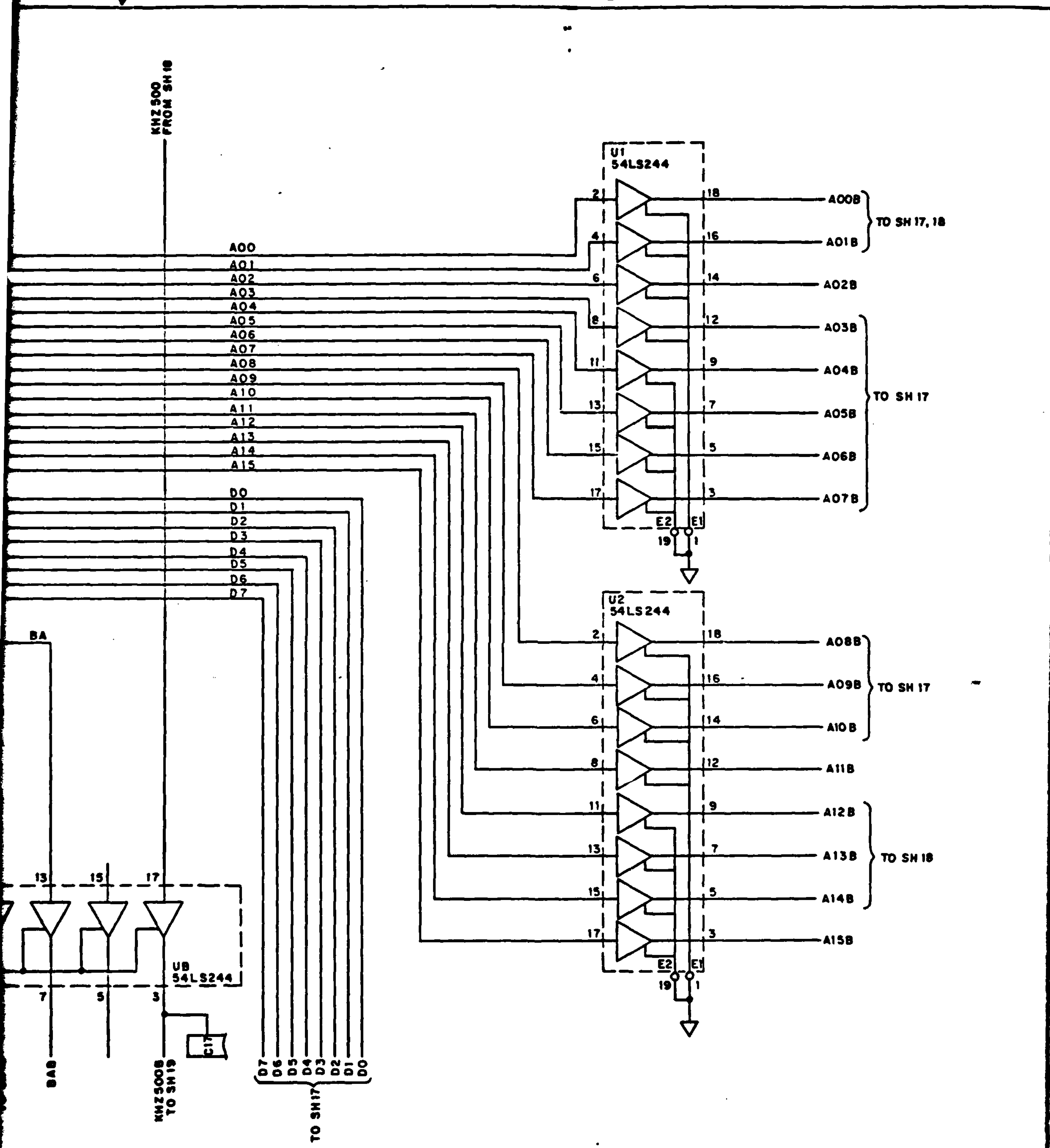
SIZE	CODE IDENT NO.	DWG NO.
D		
SCALE -	REV	SHEET 15

3 2 1

2



6
5
4
D
C
B
A
+5V
Vcc
MPU
U5/68800L
U18R3
3K
U18R2
3K
U18R1
3K
BUSΦ2
DBE
NMI-
TSC
HALT
HALT-
IRQV
B13
IRQ-
RESET-
Φ1
Φ2
U9
OSC
CO-238A-2
@4.0MHZ
MHZ4.0
X1
X2
EXT IN
MR
DMA REQ
PWR ON RST
RSTCAP
C7
6.8UF
35WVDC
MEMCLK
DMA GRANT
RSTΦ-
4xfo
2xfo
BUS Φ2
U10
6875L
BA
VMA
R/W-
Vss
U18R4
3K
RSTΦ-
U14A
54LS08
CLEAR-
A10
TCLEAR-
E1
E2
4xfo TO SH 17
CLEAR- TO SH 17,18
EΦ2B TO SH 17,18
R/W-B TO SH 17,18
VMAB TO SH 18
BAB

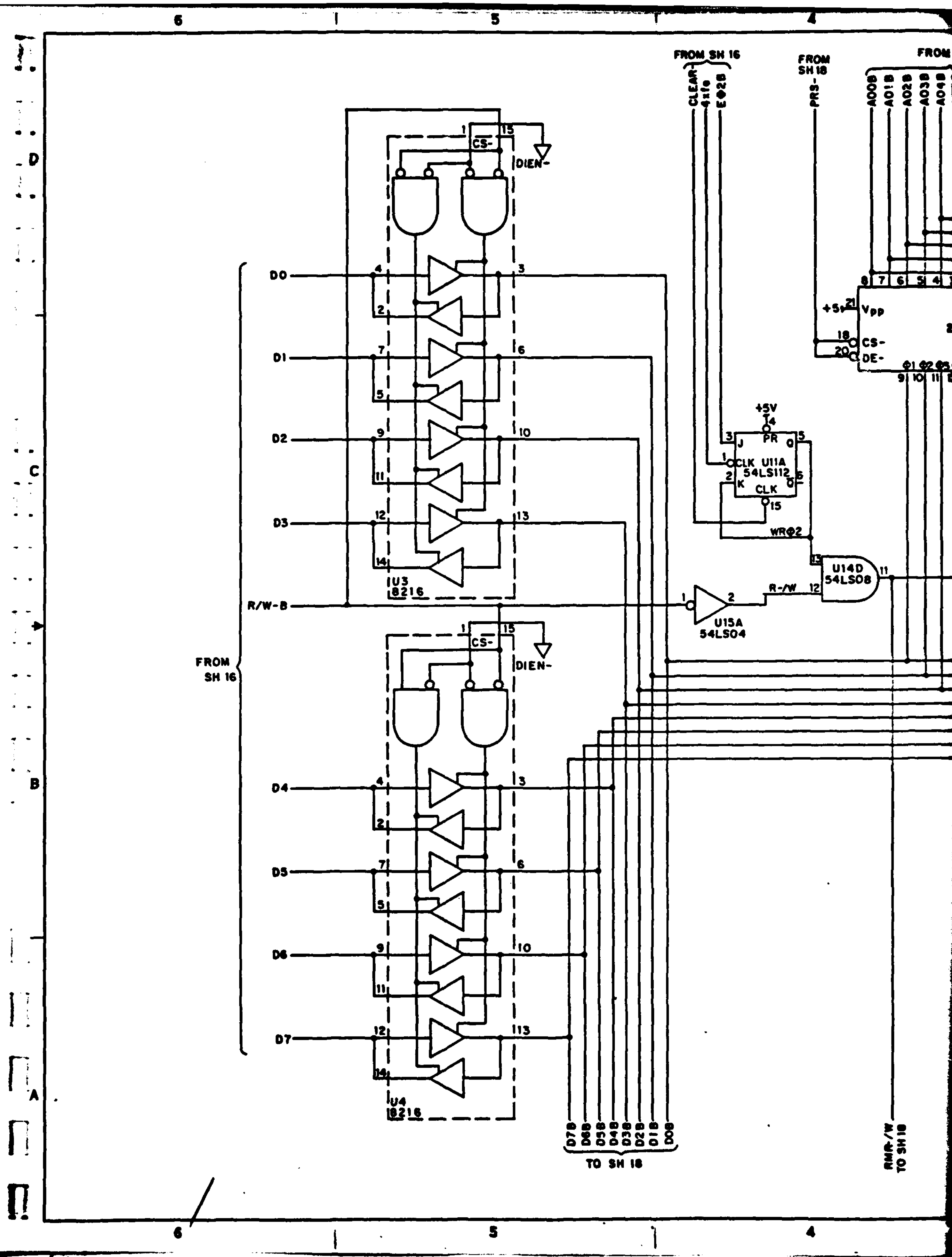


A5 F MPB

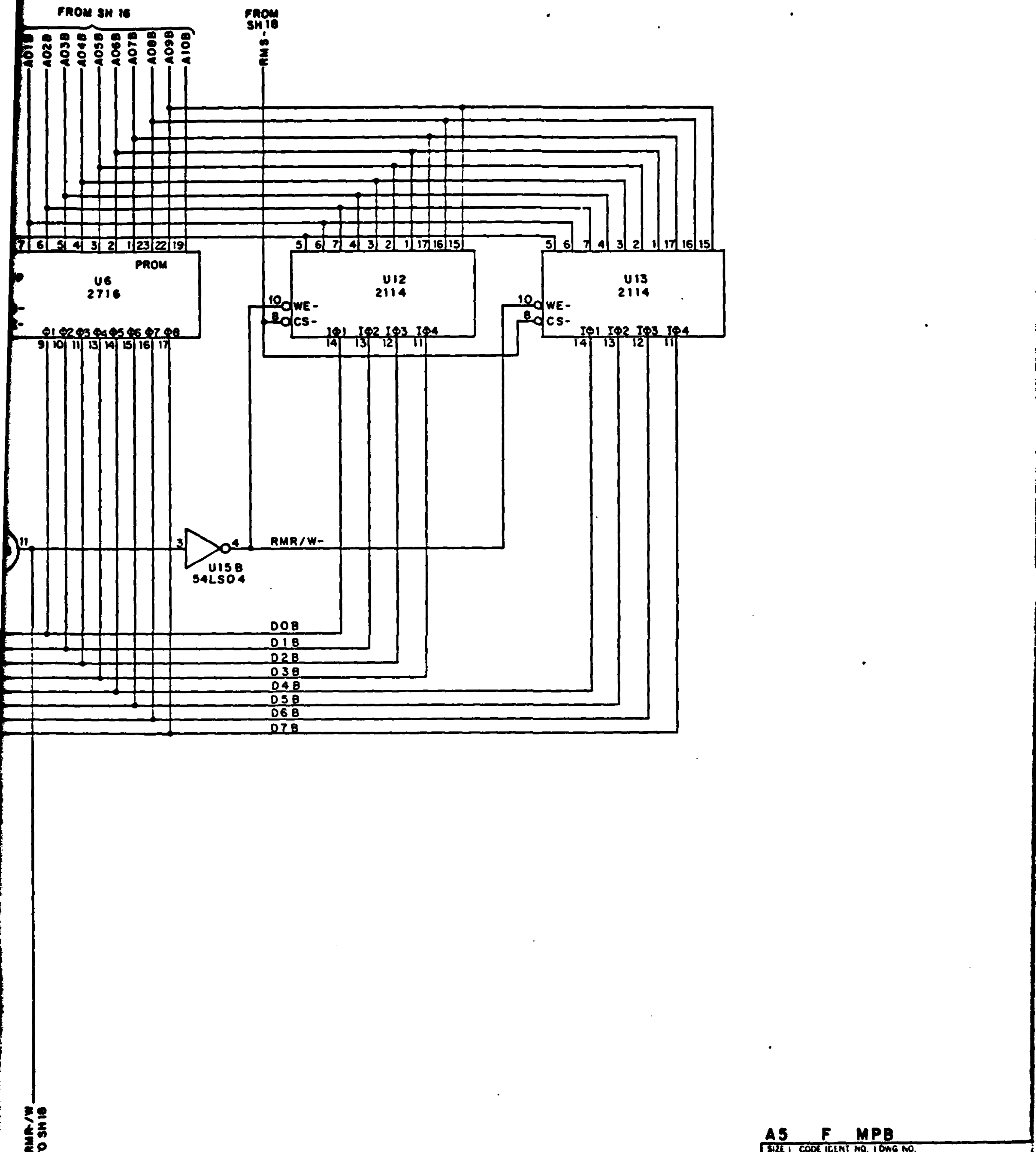
SIZE	CODE IDENT NO.	DWG NO.
D		

SCALE — REV SHEET 16

2



6
5
4
D
C
B
A
FROM SH 16
CLEAR
E02B
FROM SH 18
PRS-
FROM
A00B
A01B
A02B
A03B
A04B
CS-
DIEN-
D0
D1
D2
D3
D4
D5
D6
D7
R/W-B
U3
8216
U4
8216
FROM SH 16
+5V
J
PR
CLK
U11A
54LS112
K
WR02
U14D
54LS08
R-/W
U15A
54LS04
Vpp
CS-
DE-
D7B
D6B
D5B
D4B
D3B
D2B
D1B
D0B
TO SH 18
RMR-/W
TO SH 18

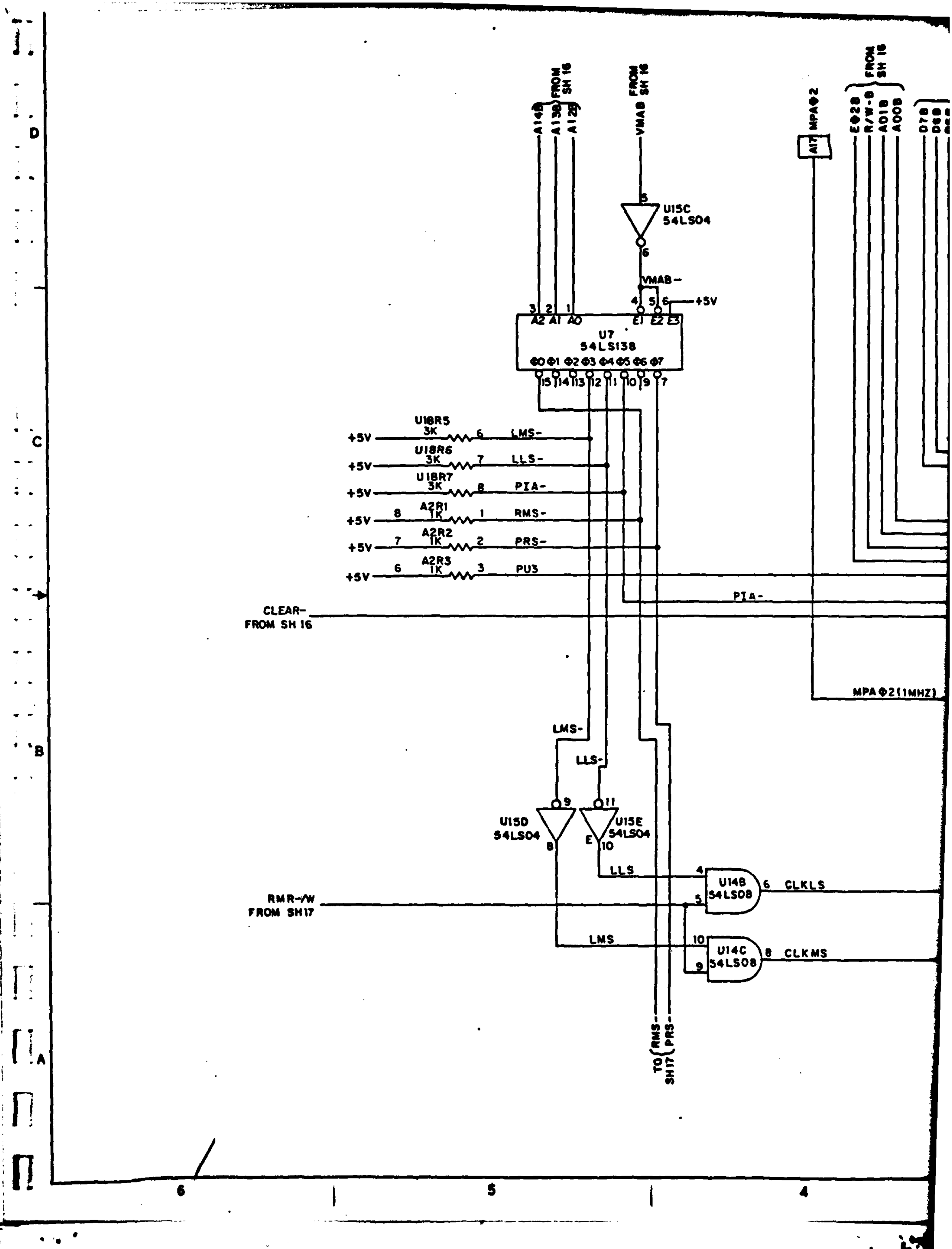


AS F MPB

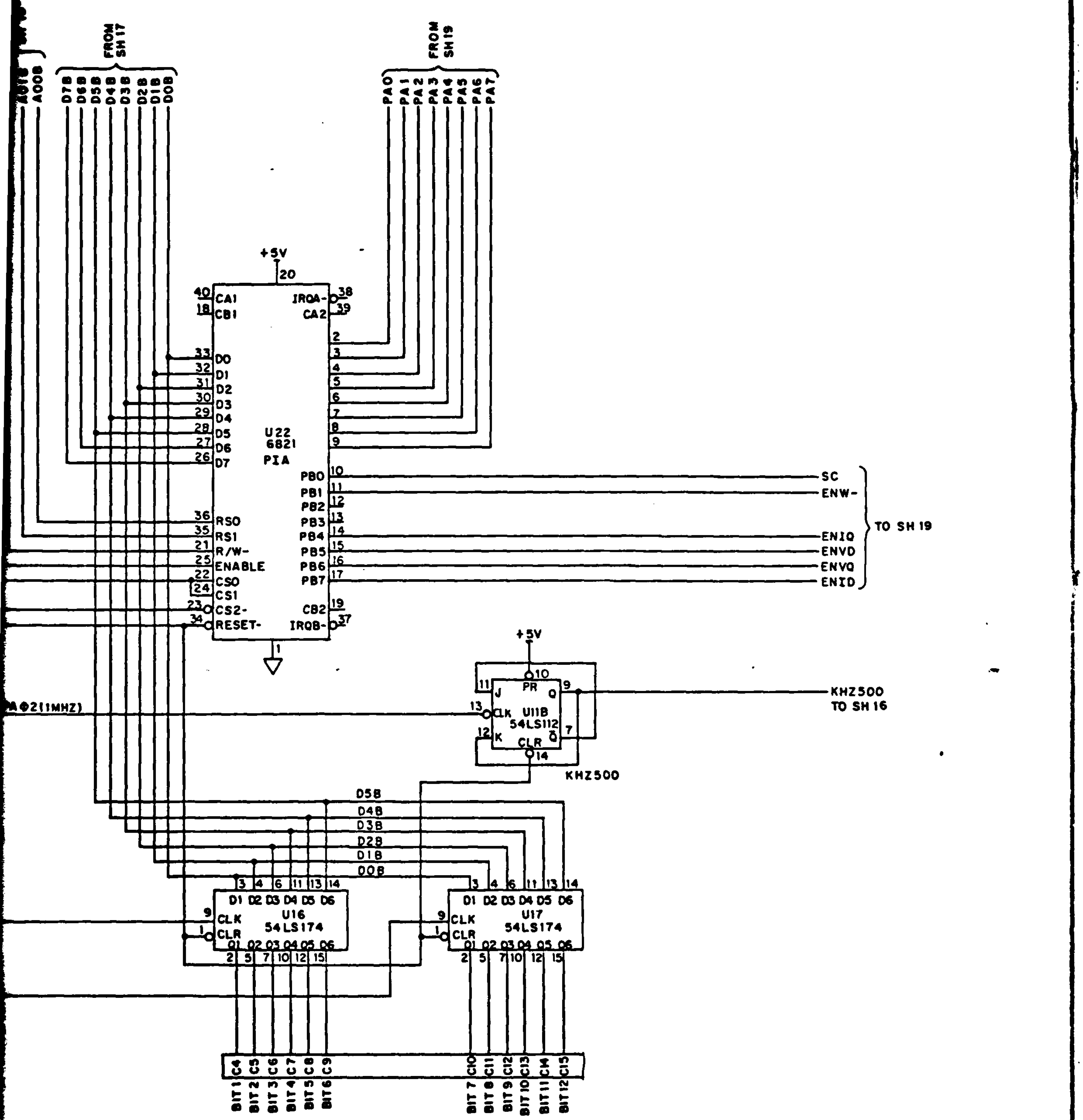
SIZE	CODE IDENT NO.	DWG NO.
D		

SCALE — REV SHEET 17

3 2 1 2



A14B
A13B
A12B
FROM SH 16
VMAB
FROM SH 16
MPAΦ2
A17
EΦ2B
R/W-B
A01B
A00B
FROM SH 16
D7B
D6B
U15C
54LS04
VMAB-
+5V
U7
54LS138
U18R5 3K
U18R6 3K
U18R7 3K
A2R1 1K
A2R2 1K
A2R3 1K
LMS-
LLS-
PIA-
RMS-
PRS-
PU3
+5V
CLEAR-
FROM SH 16
MPAΦ2(1MHZ)
U15D
54LS04
U15E
54LS04
LLS
LMS
RMR-/W
FROM SH17
U14B
54LS08
CLKLS
U14C
54LS08
CLKMS
TO SH17
RMS-
PRS-
D
C
B
A
6
5
4

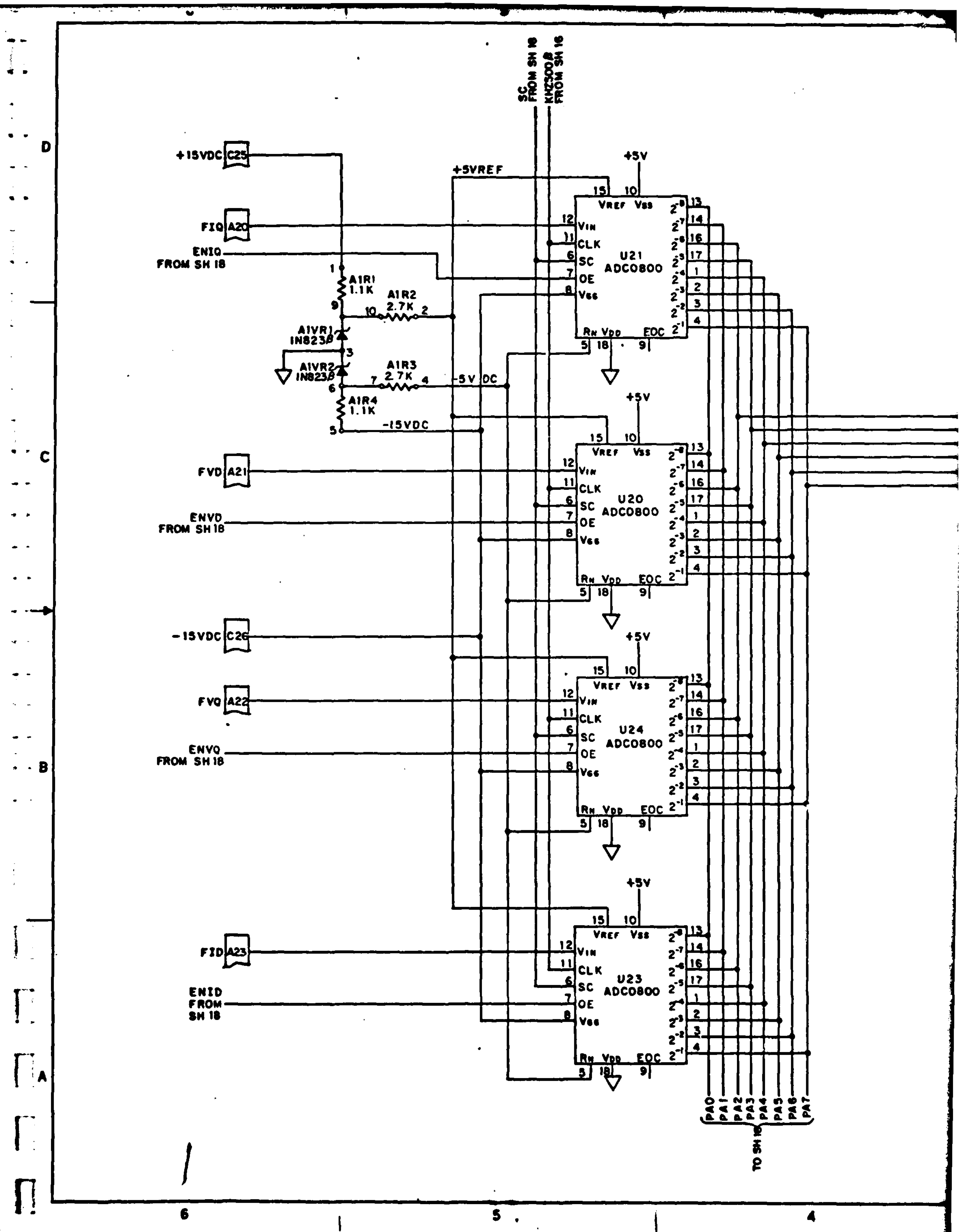


A5 F MPB

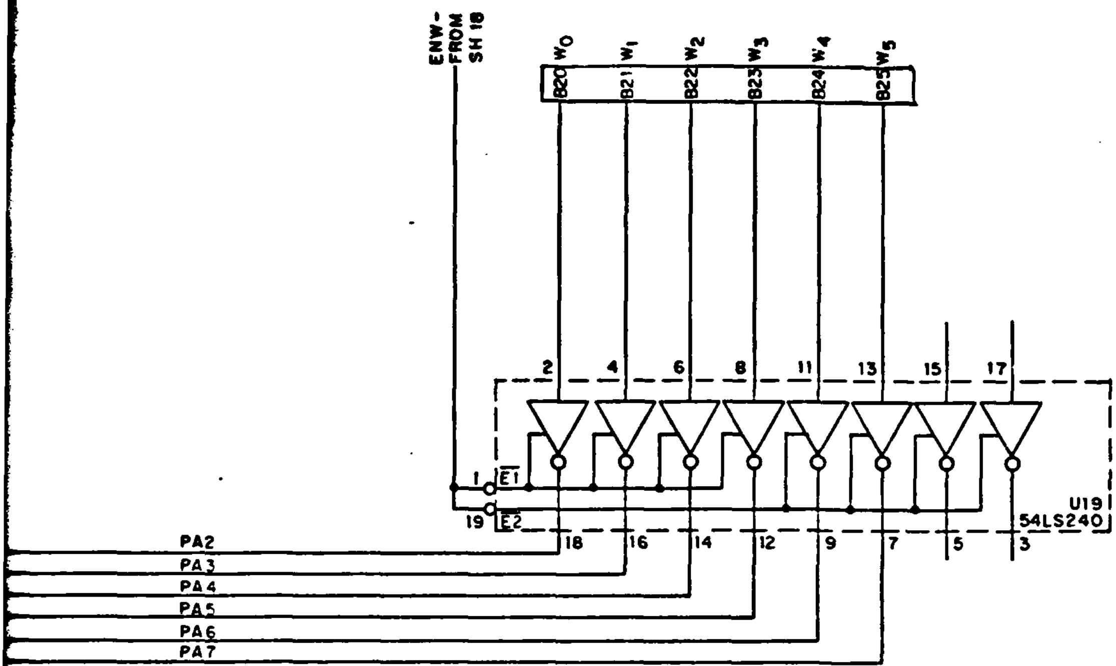
SIZE	CODE IDENT NO.	DWG NO.
D		

SCALE — REV SHEET 18

2

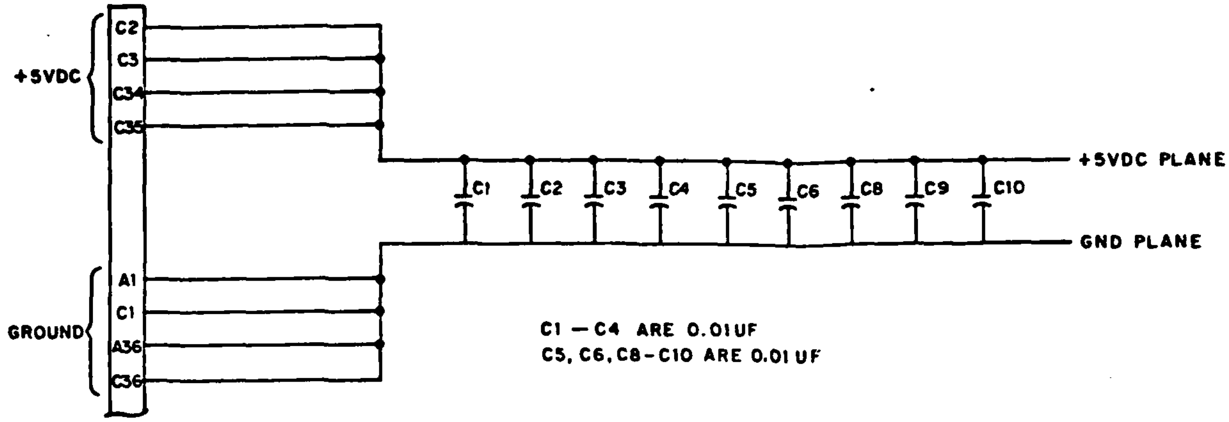


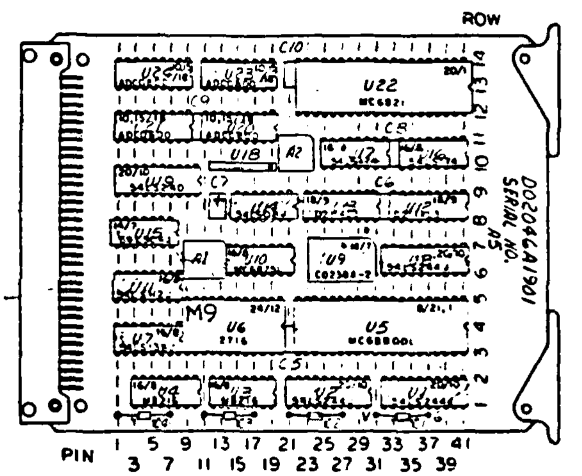
SC FROM SH 18
KHZ500B FROM SH 16
+15VDC C25
+5VREF
+5V
FIQ A20
ENIQ FROM SH 18
A1R1 1.1K
A1R2 2.7K
A1VR1 1N823B
A1VR2 1N823B
A1R3 2.7K
-5V DC
A1R4 1.1K
-15VDC
U21 ADC0800
FVD A21
ENVD FROM SH 18
U20 ADC0800
-15VDC C26
FVQ A22
ENVQ FROM SH 18
U24 ADC0800
FID A23
ENID FROM SH 18
U23 ADC0800
VREF VSS VIN CLK SC OE VGG RN VDD EOC
PA0 PA1 PA2 PA3 PA4 PA5 PA6 PA7
TO SH 18
D
C
B
A
6
5
4



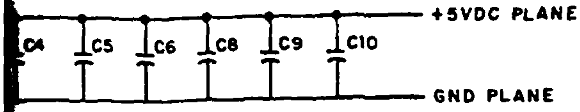
A5 F MPB

SIZE	CODE IDENT NO.	DWG NO
D		
SCALE —	REV	SHEET

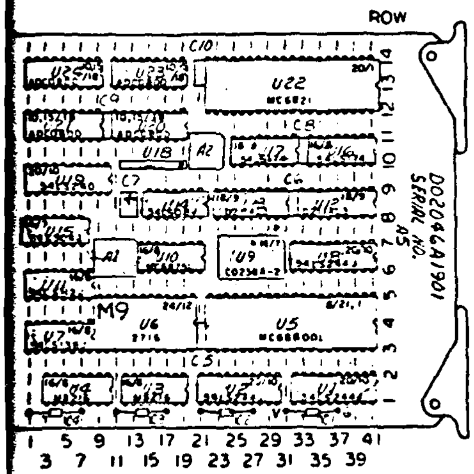

+5VDC
C2
C3
C34
C35
GROUND
A1
C1
A36
C36
C1
C2
C3
C4
C5
C6
C8
C9
C10
+5VDC PLANE
GND PLANE
C1 — C4 ARE 0.01UF
C5, C6, C8-C10 ARE 0.01 UF


ROW
D0204GA1901
SERIAL NO.
A5
U22
MC6821
U5
MC68B00L
U6
2716
M9
PIN
1 5 9 13 17 21 25 29 33 37 41
3 7 11 15 19 23 27 31 35 39
1 2 3 4 5 6 7 8 9 10 11 12 13 14

3 2 1



E 0.01UF
C10 ARE 0.01 UF



A5 F MPB

SIZE	CODE IDENT NO.	DWG NO.
D		
SCALE —	REV	SHEET 20 OF

3 2 1

Motor Controller Phase II
Task B Report

Table of Contents

I. Introduction

This Phase II, Task B report describes the results of the final of three developmental stages of the advanced motor controller. Phase I developed the approach for the digital synthesis of 3-phase, variable frequency, variable voltage, sinusoidal waveforms using an 8-bit microprocessor. The Phase I effort also analysed a method of deriving an inductive motor air-gap-flux related signal useful for optimally controlling motor voltage with varying load. Phase II, Task A developed the design for the digital and microprocessor circuitry for synthesizing the 3-phase sinusoidal waveforms and the control circuitry for deriving the air-gap-flux related feedback signal. The Phase II, Task B effort developed a 3-phase power-bridge design and breadboard, combined the power-bridge with the waveform synthesis and feedback control circuitry developed in Task A and tested the combined system.

Summary

The power-bridge and PWM control circuits were combined and tested to the design goal of 7.5 hp with excellent performance. Motor voltage control under varying load using the air-gap-flux related feedback signal was demonstrated at frequencies above 15 Hz, however, control was less than optimal. Other motor-controller features including speed ramp-down under overload conditions, power switch over-current protection, and the maintenance of constant AC output voltage under conditions of varying DC input voltage were successfully tested.

II. Power Bridge Circuit Description

The three-phase power bridge circuit consists of 6 identical power switch units connected as shown in Figure 1. The switches are driven by the control circuits to produce pulse width modulated variable-frequency, variable-voltage, three phase power. The Phase II, Task A report describes the control circuits

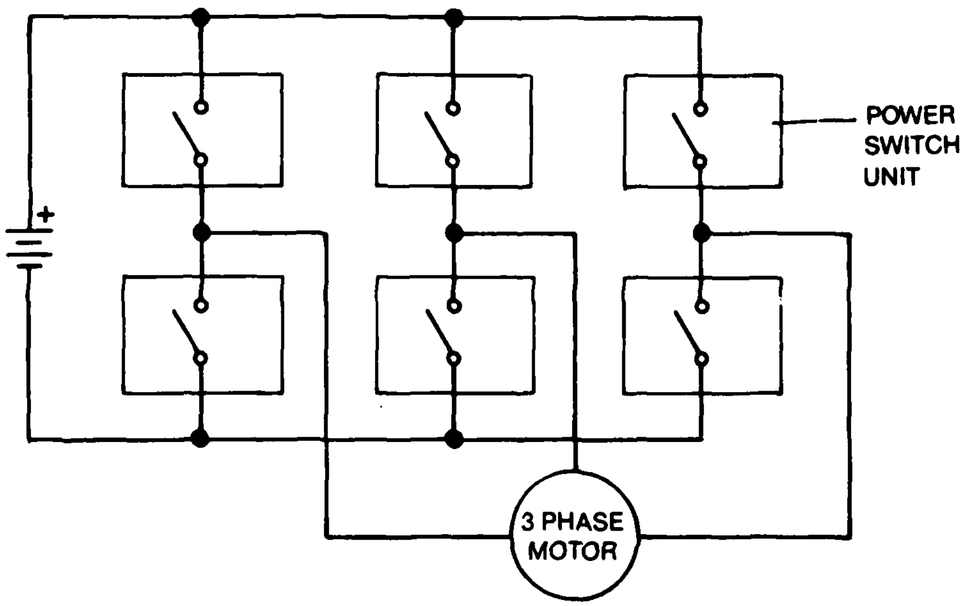


Figure 1. 3-Phase Power Bridge

and the method for generating the proper control signals for the power switches. Each of the power switches is optically isolated from the control signal source and each has individual current limit protection.

Power Switch Unit Description

The power switch units consist of a transistor switch module, a "catch" diode, a switch driver circuit, a low-voltage power supply, and a series current shunt resistor connected as shown in Figure 2.

The transistor switch module conducts the load (motor) current when in the on state and the "catch" diode conducts the load current when the switch module is off. The switch driver circuit provides the proper current pulse to drive the switch module on or off according to the control signal. The low voltage power supply provides isolated power for the switch driver circuit. The shunt provides a current proportional signal that is used for switch-module over current protection.

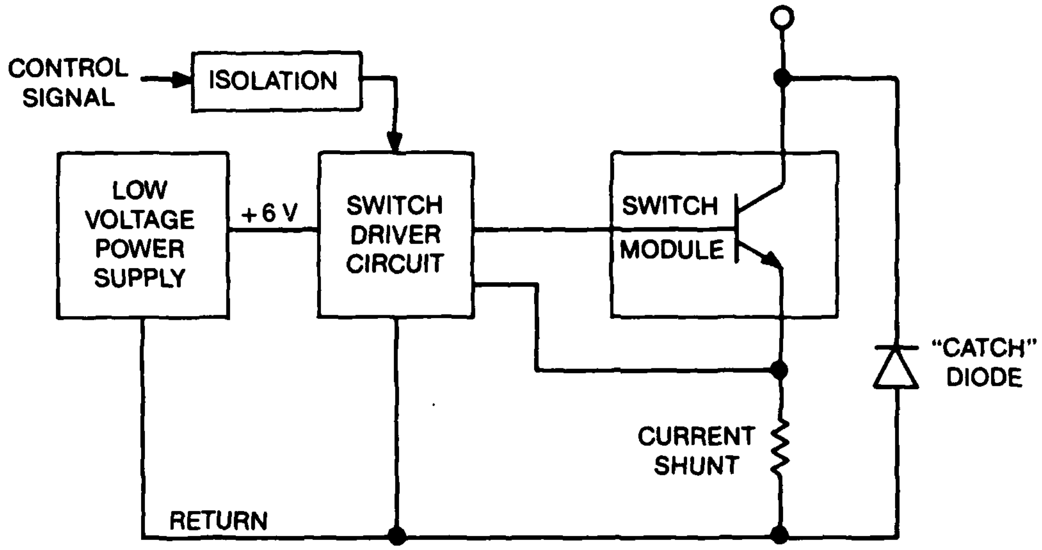


Figure 2. Power Switch Unit

Transistor Switch Module Description

The switch module consists of a matched set of transistors connected in a Darlington arrangement as shown in Figure 3. The transistors are matched by the manufacturer and housed in a 3.25 x 2.75 x 1.25 inch module. Each individual transistor is a PT-7511 (Power Tech, Inc.) (See Data Sheet, Appendix A). The switch module characteristics are as follows:

1. Collector-Emitter Breakdown Voltage BV_{CEO} = 200 min at I_C = 200 ma.
2. Emitter Cut-Off Current I_{EBO} = 15ma max at V_{EB} = 8V I_{CB} = 0.
3. Collector Cut-Off Current I_{CES} = 10ma max at V_{CB} = 200V R_{BE} = 0.
4. D.C. Current Gain hfe = 100 min at V_{CE} = 5V I_C = 250A.
5. Base-Emitter Voltage V_{BE} = 3.0V max at I_C = 250A V_{CE} = 4V.

The modules used in the motor controller were tested by the manufacturer to the specification. (See Test Data, Appendix A).

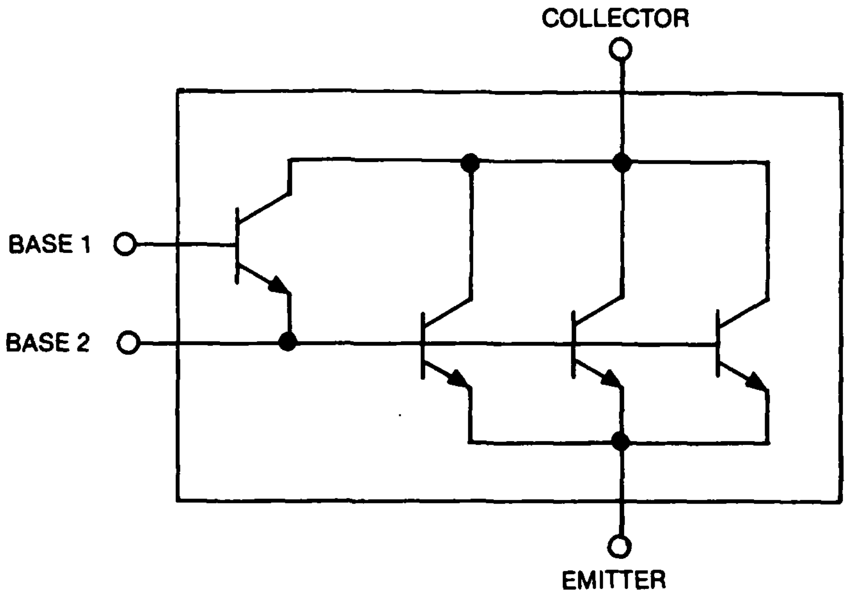


Figure 3. Switch Module

Diode Description

The "catch" diode provides a return path for the load current when the associated switch module is turned off. The manufacturer is Semtech Corp., part numbe SCSF4R. The important characteristics are as follows:

Peak Inverse Voltage	400V
Forward Voltage	$V_f = 1.40$ max at $I_f = 100A$
Recovery Time	$t_{rr} = 200$ nanoseconds (measured when rectifier recovers to 0.25 amp from a 0.5 amp forward current)

Reverse polarity diodes were used in the breadboard such that the cathode could be connected to the collector of its associated switch module using the heat sink as the conductor.

Switch Driver Circuit Description

The switch driver circuit inputs a control signal from the control electronics and outputs corresponding turn-on and turn-off drives to the base of its associated transistor switch module. A block diagram of the circuit is shown in Figure 4. The circuitry represented by the blocks to the left of the vertical dotted line in Figure 4 is located on a 4 x 4 inch Printed Wiring Assembly and the remaining circuitry is located on the switch module heat sink assembly. The circuitry operates as follows: (See Driver Board Schematic Diagram, Appendix B).

1. Input Isolation

Input isolation consists of an optical isolator, U1, that receives a TTL logic signal from the control circuit. A logic "0" turns the switch module on and a logic "1" turns it off. The optical isolator permits the power switch unit and the control circuit to operate at different reference voltages.

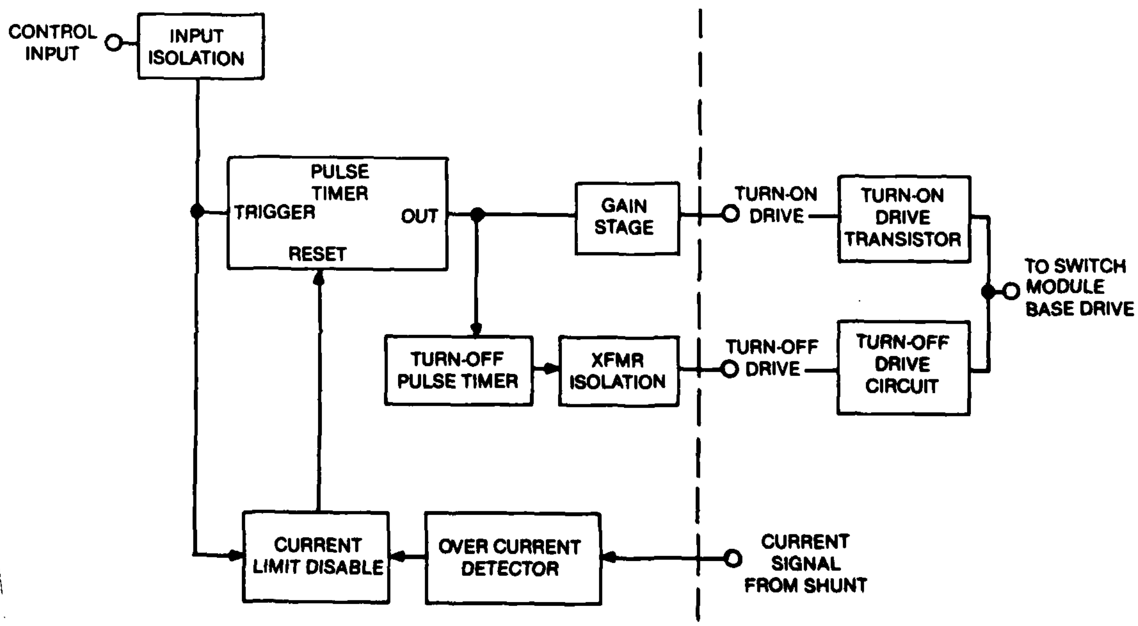


Figure 4. Switch Driver Block Diagram

2. On-Drive Timer

The pulse on-drive timer, U2A, is a protective circuit that limits the maximum on-time of the switch module. The timer turns off the switch module should the control circuit fail to do so within the prescribed time. The timer is triggered on by a logic "0" and reset by a logic "1" on the control input. The timer can also be reset by a signal from the overcurrent protection circuit.

3. Current Limit Disable

The current limit disable circuit disables the overcurrent protection circuit for the initial 6 microseconds of each power switch turn-on cycle. The circuit prevents the detection of overcurrents due to the recovery current of the "catch" diode or L di/dt voltages in the shunt circuit that normally occur when a switch is turned-on. The overcurrent-disable time period is established by comparing the turn-on pulse from the input optical isolator, differentiated by C2 and R7, with a reference voltage generated by R19 and VR1. The output of the comparator U3A is wired in an "OR" arrangement with the current limit comparator U3B such that the overcurrent limit is disabled. The waveforms are diagrammed in Figure 5.

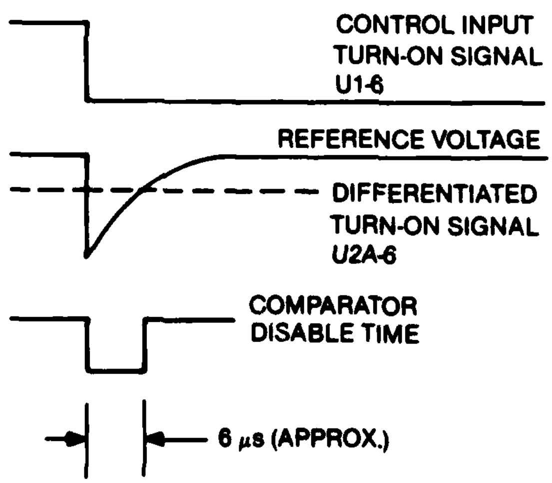


Figure 5. Current Disable Waveforms

4. Overcurrent Detector

The overcurrent detector compares the switch module current with a reference and resets the pulse time if the reference is exceeded. The voltage across the current shunt is sensed and amplified by a differential op amp, U6A, such that common mode rejection of the op amp reduces the effects of the L di/dt induced voltage in the shunt circuit. (See schematic diagram, Appendix D). The current signal is compared with a reference, U6B, and if the reference is exceeded the pulse timer is reset through U3B. The shunt resistance is 0.002Ω and the differential op amp gain is 5 such that the reference voltage (pin 5 of U6) may be established by the following formula:

$$V = 0.01\ I_{LIMIT} + 0.7$$

The reference is presently set at 2.00 volts for current limiting at approximately 125 amps.

5. Off-Drive Timer and Isolation Transformer

The overall purpose of the turn-off circuit is to minimize the turn-off time of the transistor switch modules by reverse biasing the base-emitter junctions to rapidly remove the stored charge in the base regions of the transistors. The off-drive timer supplies a 40 μs pulse that establishes the period during which the reverse voltage is applied to the switch module base-emitter junction. The timer is triggered by the termination of the turn-on drive and its output drives the primary winding or an isolation transformer T1. The isolated pulse from the secondary winding of T1 drives a second transformer (See Power Stage Schematic Diagram, Appendix B) that generates the reverse voltage. The actual base drive pulse is shown in photos 1 and 2.

Low Voltage Power Supply

Each of the 6 power switch units has an associated low-voltage power supply for powering the switch driver circuits. The supplies are standard commercial AC to DC types rated at 5VDC 3 amps.

Shunt Resistor

The shunt resistor conducts the power switch emitter current and is used for current sensing. The shunts are made of strips of monel metal cut to dimensions that produce a 0.002Ω resistance between terminals. The voltage generated across the shunts is connected to the differential op amp in the overcurrent detector circuit. Monel is used because of its low temperature coefficient.

Hardware Protective Circuits

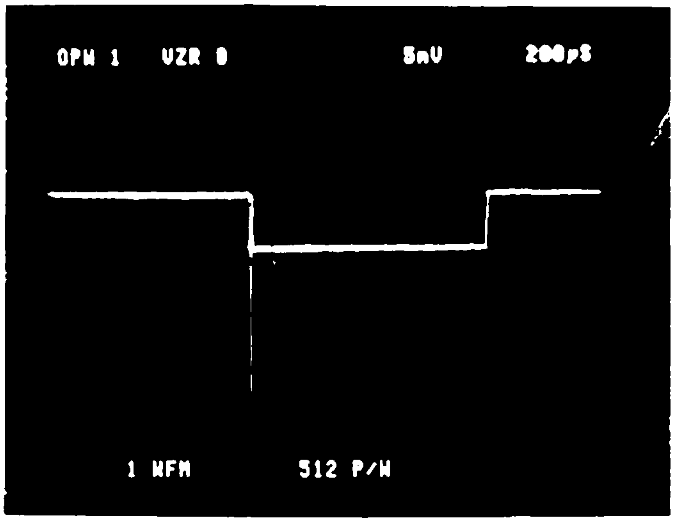
Each power switch unit has been designed to turn-on only from a command from the control circuits but will turn off under three conditions: (1) a control circuit command; (2) an overcurrent condition; or (3) in the event of no control circuit turn-off command within a prescribed period. Refer to the Switch Driver circuit description for details in the operation of these circuits. In addition, shoot-through protection has been designed in the control circuit such that a 26 microsecond off-period occurs between the time a power switch unit turns-off and its complementary power switch turns-on. This off-period insures that the conducting periods for complementary power switch units do not overlap.

III. Power Bridge Test Results

The Motor Controller Breadboard is shown in Photo 3 and the motor and dynamometer used for test purposes shown in Photo 4. The motor is a Franklin, oil-cooled type, rated at 7.5 hp, 600 rpm, 87V, 32 Hz. The dynamometer is rated at up to 125 hp. The power bridge was tested over the range of frequencies 6 to 60 Hz at various loads and was specifically tested to the motor ratings of 7.5 hp, 600 Rpm, 32 Hz.

Performance of the power bridge was excellent in all respects. Photos 5 through 12 show the phase to phase voltage and phase current waveforms under medium load at 15, 30, 45, and 60 Hz. Photo 13 shows the collector-emitter voltage of one of the switch modules and photo 14 shows the waveform of the current into one power switch unit including the "catch" diode current. Photo 15 shows the unfiltered phase to phase voltage for a high output voltage (wide pulse width) and photo 16 shows the same waveform for a low output voltage (narrow pulse width).

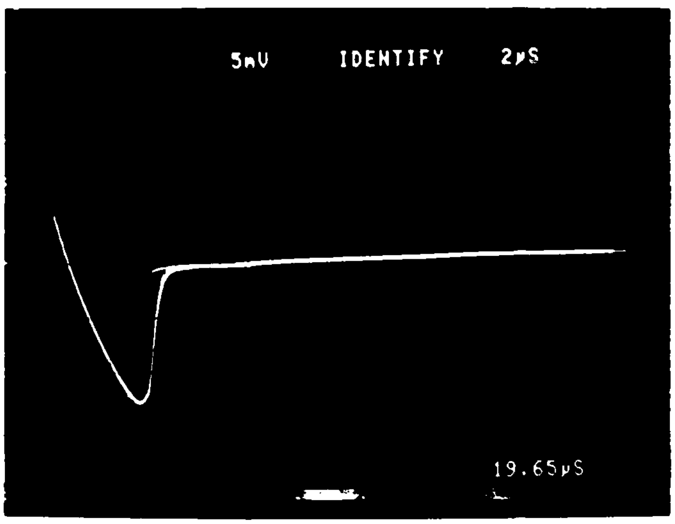
The switching waveform for the power switch modules were such that snubber networks were not required. The improvement in switching waveform would not compensate for the additional power dissipation and circuit complexity of snubber networks. This was a result of two design decisions: (a) to use relatively slow power transistors; and (2) to use a careful layout of the bus bars and filter capacitor to insure clamping of the bus voltage, in conjunction with fast "catch" diodes.



1

ON OFF

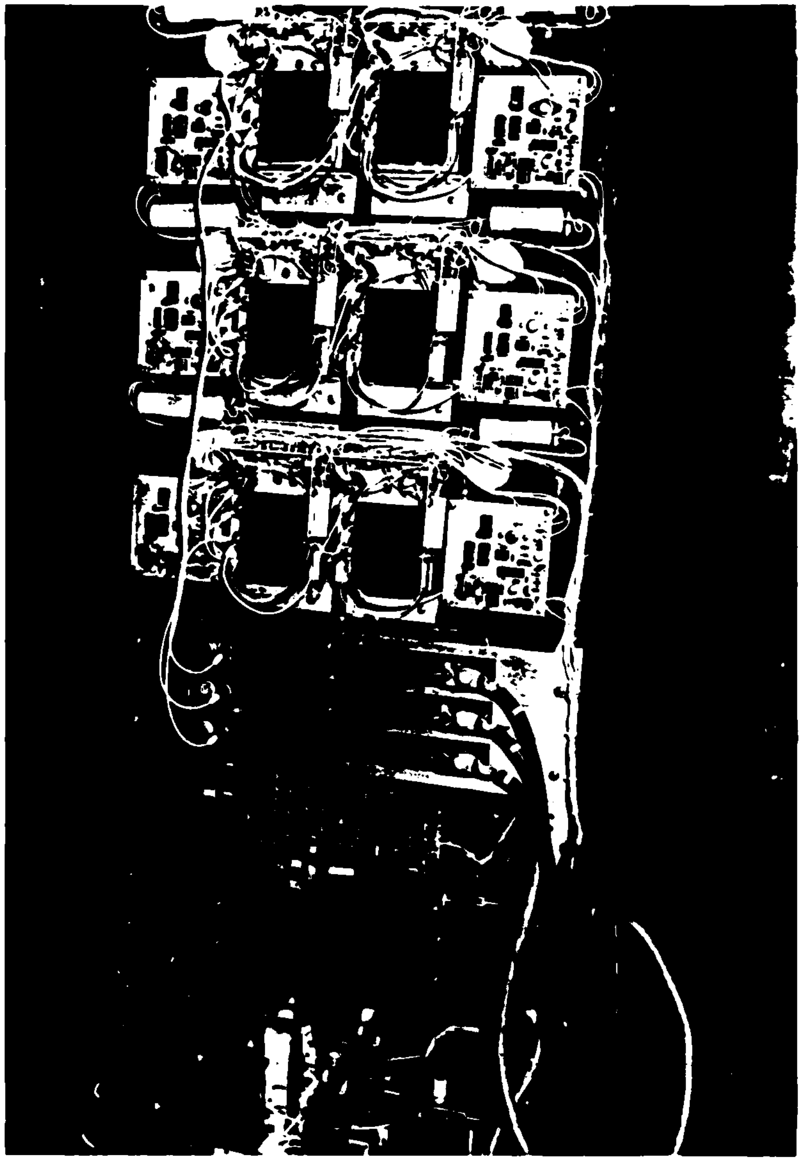
Switch Module Base Drive Current
2 Amp/Div



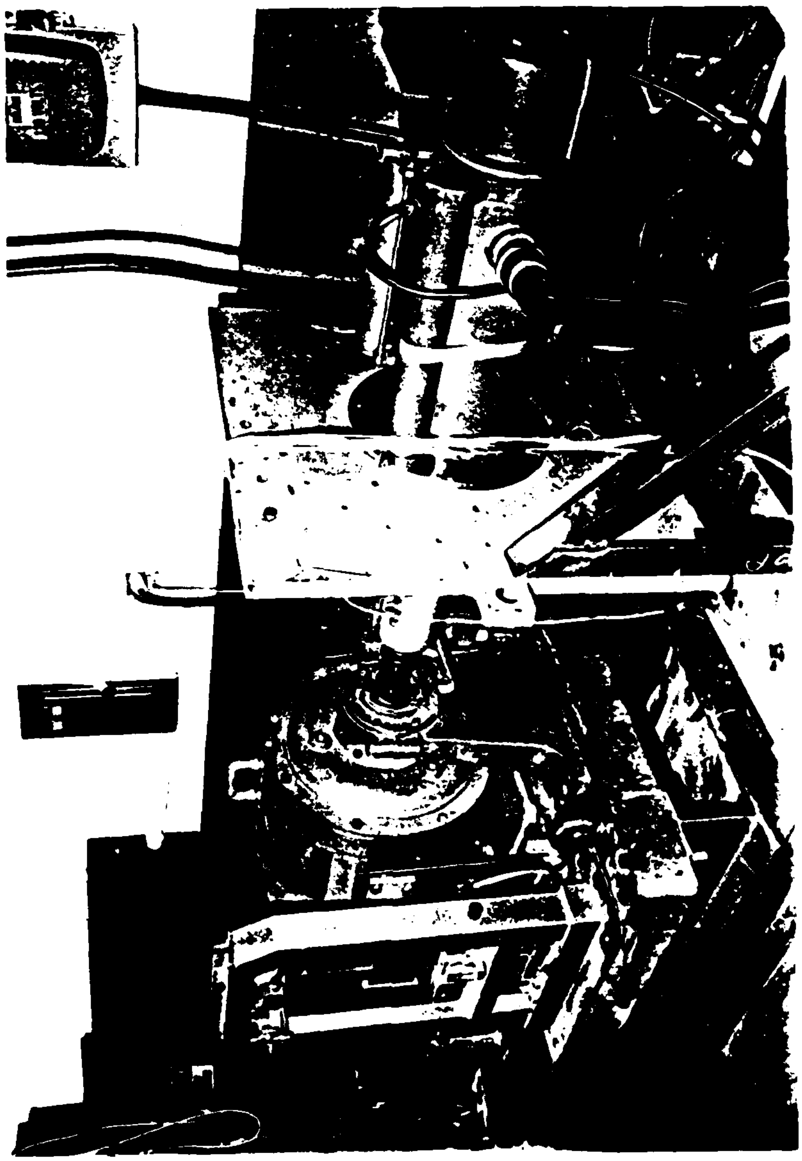
2

Switch Module Base Drive Expanded to
Show Turn Off Reverse Current
2 Amp/Div

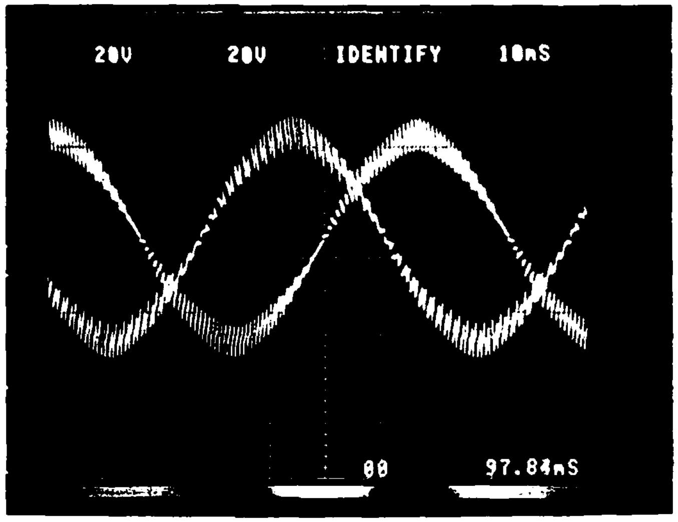
3



Motor Controller Breadboard

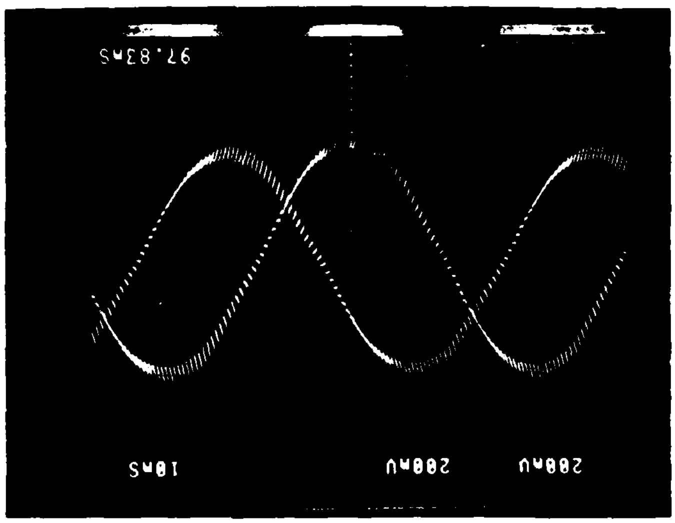


Motor and Dynamometer



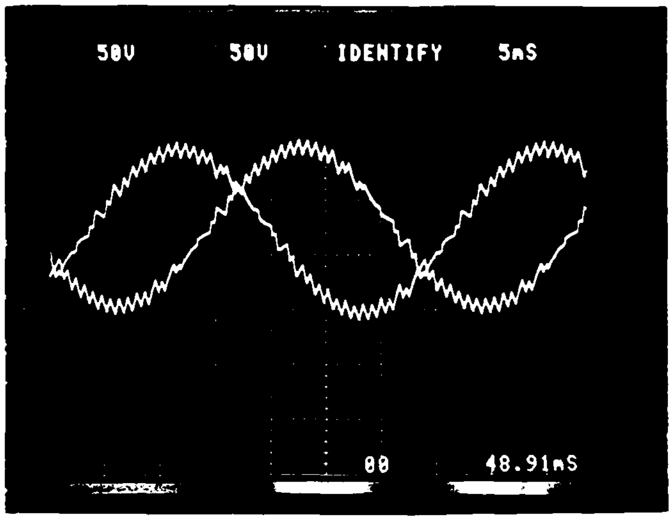
5

ϕ - ϕ Voltage (2 Phase) 15 Hz (20 V/Div)
Filtered (100 Hz Cutoff)



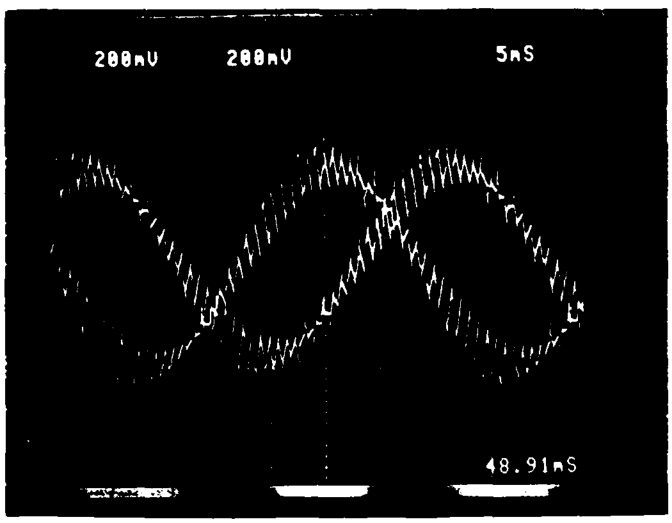
6

ϕ Current (2 Phase) 15 Hz (20 A/Div)
Unfiltered



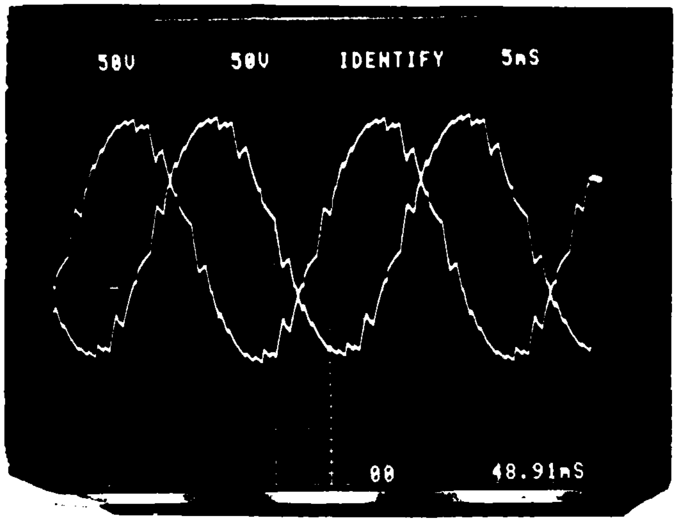
7

ø - ø Voltage (2 Phase) 30 Hz (50 V/Div)
Filtered (100 Hz Cutoff)



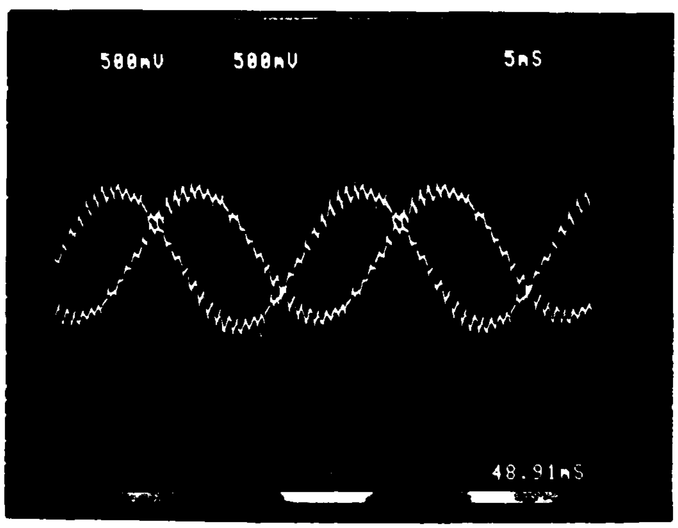
8

ø Current (2 Phases) 30 Hz (20 A/Div)
Unfiltered



9

Ø - Ø Voltage (2 Phase) 45 Hz (50 V/Div)
(Filtered — Hz Cutoff)

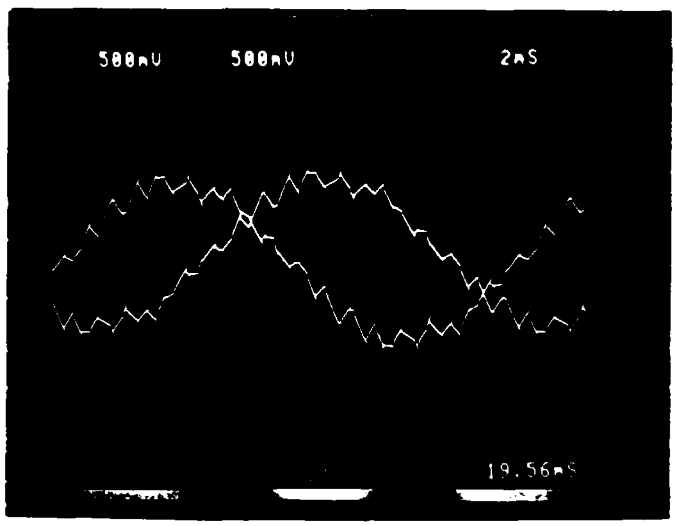


10

ØCurrent (2 Phases) 45 Hz (50 A/Div)

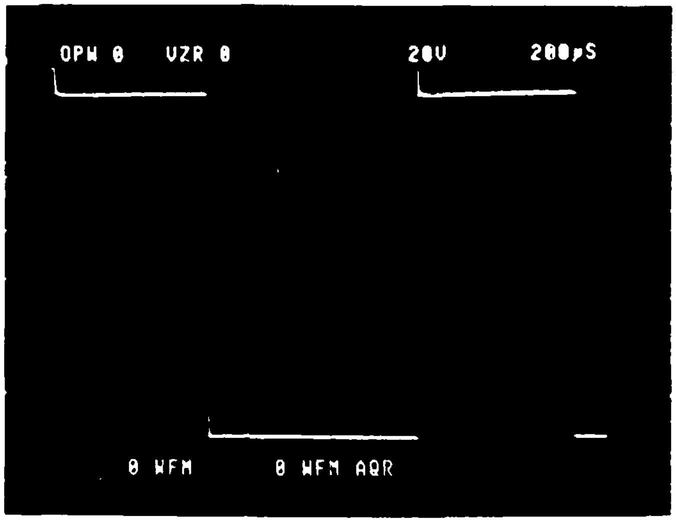
11

Ø - Ø Voltage (2 Phase) 60 Hz (50 V/Div)
(Filtered — Hz Cutoff)



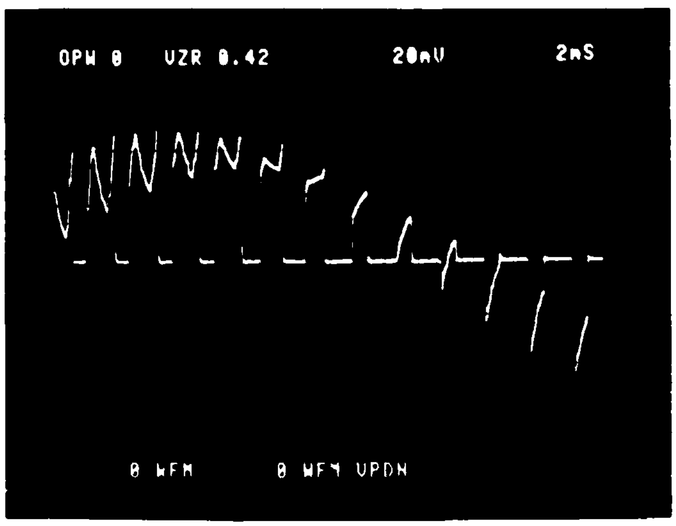
12

Ø Current (2 Phase) 60 Hz (50 A/Div)



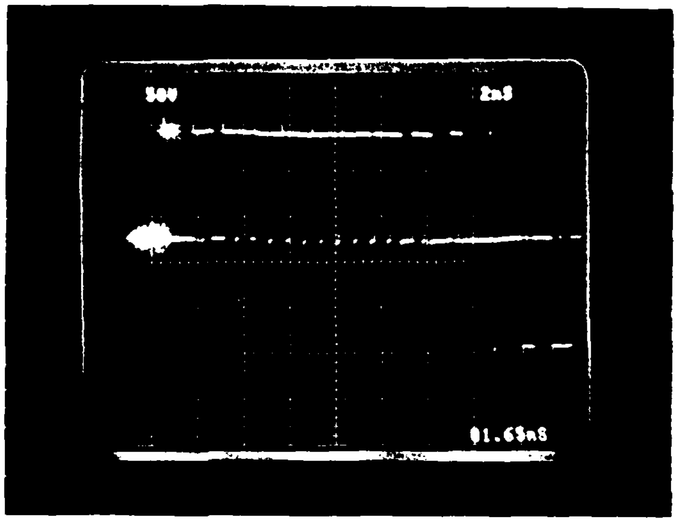
13

Vce (One Switch Module) (20 V/Div)



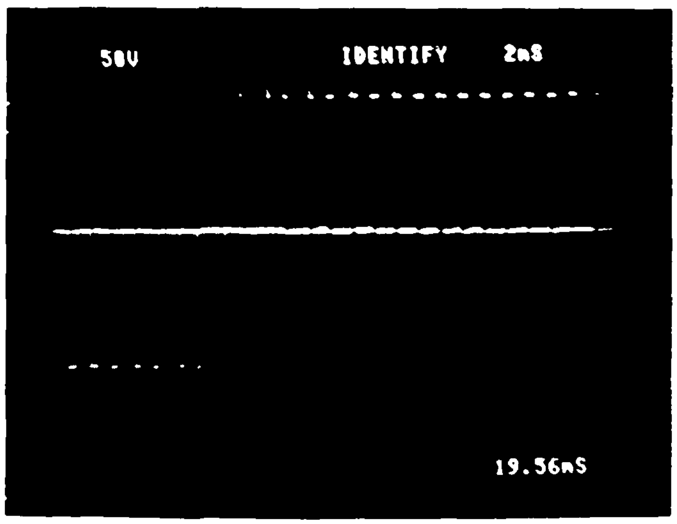
14

Power Switch Unit Collector Current Including "Catch" Diode Current



15

Unfiltered ϕ - ϕ Voltage (30 Hz) High Output Voltage



16

Unfiltered ø - ø Voltage (30 Hz) Low Output Voltage

IV. Motor Controller Test Results

The power-bridge and feedback control circuits were connected to test the hardware and software required for motor-controller closed-loop operation. The resistance and inductance values for the test motor were provided by NSRDC as follows:

$R_S = 0.0605\Omega$

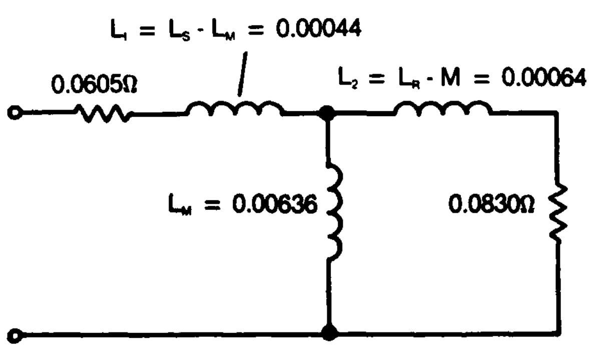
$L_S = 0.00680h$

$R_R = 0.0830\Omega$

$L_R = 0.00700h$

$M = 0.00636h$

The equivalent circuit values were derived as follows:



The control circuits and software were tailored to the test motor as follows:

(1) The stator inductance value L (Refer to Phase I report) used for the feedback equation was calculated as follows:

$$L' = L_1 + L_M L_2/(L_M + 2L_2)$$

$$L' = 0.00044 + (0.00636)(0.00064)/[0.00636 + (2)(0.00064)] = 9.73 \times 10^{-4} h$$

A table of K_1 values was computed and entered into EPROM memory as follows:

$$K_1 = 2\Pi f L' \times \text{scaling factor}$$

A single K_1 value (16 bits) for each integer frequency from 6 to 60 Hz was entered into the table. (PWM microprocessor program location FE06 Hex).

(2) The direct and quadrature currents and voltages were scaled by installing the appropriate resistors in the scaling circuits located on the A4 PWA in the control electronics. Scaling is as follows:

V_d and V_q scaled for	± 128V peak
I_d and I_q scaled for	± 256 amps peak

(3) The nominal (open-loop) output voltage table (Vopen) was established for the test motor. The no-load motor voltage at which the motor begins to stall was measured at several excitation frequencies. A linear V/F = table for the full range of frequencies was established from these measurements and the table was entered into the PWM processor memory.

An initial attempt to close the air gap flux feedback loop was partially successful. For excitation frequencies above 15 Hz at light to medium motor loads the feedback loop varied the motor voltage proprotionally with load variations as expected. At high loads the feedback error calculated in software exceeded the maximum allowed by the software filter subroutine and an overflow occured. At low excitation frequencies 6 to 15 Hz the feedback loop did not correctly respond to the load variations.

Several minor changes were made to the feedback processor software to eliminate the overflow problems. The software flow was rearranged and a variable gain (table loop-up) algorithm was added to the feedback error signal, E*, as shown in Figure 6. Implementation of the selectable gain algorithm on the feedback error signal resulted in a smaller magnitude error signal in the filter subroutine, eliminating the overflow problem. The variable gain feature also added flexibility in establishing the optimal voltage boost for a given feedback error.

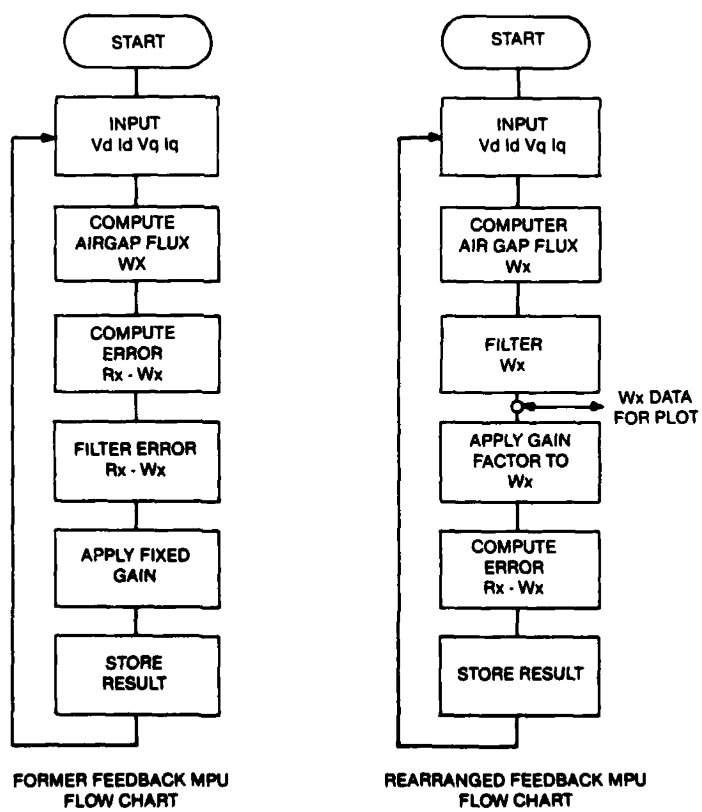


Figure 6. Revision to Feedback Flowchart

Closed-loop testing with these software modifications enabled adequate operation at excitation frequencies above 15 Hz. The motor voltage responded proportionally with load variations over the entire load range. Motor voltage regulation for a given load, however, tended to be on the underexcited side of the optimal motor excitation. Attempts at achieving optimal motor excitation voltage resulted in a runaway condition, where the excitation voltage increased to its maximum regardless of the load. A set of data was taken to examine the flux feedback signal at various excitation frequencies from 6 to 60 Hz and various motor loads. The data is tabulated and plotted in Figures7 through 12. The air gap flux related signal, W_x was recorded from feedback microprocessor program location "FABE" hex.

The feedback control loop as presently implemented requires the W_x versus motor voltage plot to have a negative slope. Regulation occurs in this negative slope area to the left of the minimum "valley" W_x value shown on the W_x plot. The calculated W_x curves for frequencies above 15 Hz have the necessary negative slope areas that allow the loop to regulate. Below 15 Hz, the curves do not have the required shape. Regulation was attained at frequencies above 15 Hz and the motor voltage increased with load as expected. Regulation was not achieved at excitation frequencies below 15 Hz. In all cases, the motor was slightly under-excitated during closed-loop regulation. The approximate optimal regulation points on the W_x versus Motor Voltage plots are shown by a small circle on each of the plots. These points were estimated from the attached data through an interpretation of the slip speed and input power. Photo 17 through 24 show the filtered feedback signals used to calculate W_x. Pictures of V_d, V_q, I_d, and I_q at 4 frequencies are shown. In all cases the "direct" quantity leads the "quadrature" quantity. The waveform photos were taken at the A/D Converter inputs in the control circuit.

Problems that affected proper motor voltage regulation using the air-gap-flux related signal included: (1) obvious distortion in the filtered Id, Iq, Vd, and Vq signals at low frequencies (see photos 23 and 24); (2) general difficult with the complexity of tailoring the feedback equation to the test motor parameters; and (3) control problems related to the double valued nature of the air gap flux related feedback signal.

In order to test the various features of the motor controller without relying on the Wx error quantity, the motor controller software was modified to provide a constant voltage per frequency output in the closed-loop mode. This change involved installing a new open-loop voltage table in the PWM processor software and using a constant for determining the PWM Scale factor rather than the error signal from the feedback microprocessor. (Constant is located at DD46 and DD47 Hex). The feedback microprocessor software was also rearranged slightly to rectify and filter the motor direct current value previously used in the air gap flux calculation and pass the result to the PWM microprocessor. The rectified current value is used by the PWM processor for software overload protection in the same manner as the Wx error signal was previously used. Software listings containing these changes are provided.

The overload, ramp-down feature was tested and the overload threshold set to approximately 100 amps peak motor current. Motor controller operation using toggle switches for motor speed changing and direction reversing rather than using the CRT Terminal was also demonstrated.

WORK SHEET
WESTINGHOUSE FORM 8295 E

OUTPUT		INPUT		W_x at "FABE"		MOTOR	DYNAMOMETER FLD. EXCIT.	6 HZ DATA
V_{AC} PEAK	I_{AC} PEAK	V_{DC}	I_{DC} AMPS	HEX	DEC	RPM	%	
25	45	125.6	3.4	0750		118	2	LIGHT LOAD
20	37	125.6	2.4	0490		117	2	
17.5	32	125.7	2.0	0329		116	2	
15	27	125.8	1.8	0241		114	2	
12.5	25	125.8	1.6	0196		112	2	
10	22	125.8	1.4	0118		107	2	
7.5	20	125.9	1.2	00A3		091	2	
6	21	125.9	1.2	0085		044	2	
30	60	125.7	5.8	0BD0		116	6	MEDIUM LOAD
25	47.5	125.7	4.4	0766		114	6	
20	40	125.3	3.6	0480		111	6	
17.5	37.5	125.3	3.2	0380		108	6	
15	40	125.4	3.0	0296		103	6	
12.5	37.5	125.4	2.8	01C0		091	6	
10	35	125.5	2.6	0160		060	6	
30	65	124.8	8.0	0C40		110	10	HEAVY LOAD
25	55	124.8	6.4	0785		107	10	
20	50	124.9	5.8	04E7		c98	10	
17.5	50	124.9	5.4	03D0		087	10	
15	50	124.9	5.2	0320		070	10	
12.5	50	125.1	4.4	0286		040	10	

FIGURE 7A

WORK SHEET
WESTINGHOUSE FORM 8295 E

10 Hz DATA

OUTPUT		INPUT		Wx at 'FABE'		MOTOR	DYNAMOMETER FLD EXCIT.	
Vac PEAK	Iac PEAK	Vdc	Idc AMPS	HEX	DEC	RPM	%	
50	80	124.7	8.6	1F9A		198	2	LIGHT LOAD
40	50	125.0	4.6	0E90		198	2	
30	37.5	125.2	3.2	0730		195	2	
25	32	125.4	2.6	04A0		193	2	
20	27.5	125.4	2.2	0300		190	2	
17.5	26	125.5	2.2	026A		187	2	
15	25	125.5	2.0	01EB		181	2	
12.5	26	125.6	2.0	0180		165	2	
10	27.5	125.6	2.0	0186		119	2	
7.5	26	125.6	1.4	0125		STALL	2	
40	52	124.8	6.8	0E38		193	6	MEDIUM LOAD
30	45	124.9	5.6	07CC		186	6	
25	42	124.9	5.2	0567		177	6	
20	47.5	124.9	5.2	044C		162	6	
17.5	47.5	124.9	5.2	0425		135	6	
15	50	125.0	5.0	0445		98	6	
12.5	47.5	125.1	3.8	038F		48	6	
50	75	123.8	15.4	1890		186	10	HEAVY LOAD
40	65	124.0	12.2	0EBA		179	10	
30	67	124.0	11.6	0A3D		158	10	
27.5	72	124.0	11.6	0900		135	10	
25	75	124.0	11.4	090F		118	10	
20	70	124.3	8.8	086E		59	10	

FIGURE 7B

WORK SHEET
WESTINGHOUSE FORM 8295 E

20 Hz DATA

OUTPUT		INPUT		W_x at "FABE"		MOTOR	DYNAMOMETER FLD. EXCIT.	
V_{AC} PEAK	I_{AC} PEAK	V_{DC}	I_{DC} AMPS	HEX	DEC	RPM	%	
60	44	124.9	6.0	139B		394	2	LIGHT LOAD
50	37.5	124.9	5.2	0C90		392	2	
40	35	125.0	4.6	0824		386	2	
30	35	125.0	4.4	0560		375	2	
25	35	125.1	4.4	0485		356	2	
20	42	125.0	4.6	05D0		293	2	
17.5	45	125.1	4.4	0790		152	2	
60	55	123.9	14.2	14C8		378	6	MEDIUM LOAD
50	60	123.8	14.0	1115		366	6	
40	65	123.7	14.2	1097		338	6	
37.5	75	123.6	14.8	1400		290	6	
35	85	123.7	14.2	18E0		175	6	
80	82	122.0	31.2	28DA		370	10	HEAVY LOAD
70	82	122.0	31.0	2660		362	10	
68	85	121.8	31.8	2590		354	10	
65	87.5	121.9	31.8	2560		352	10	
60	95	121.8	32.2	2490		337	10	
57.5	100	121.8	32.2	2770		318	10	

FIGURE 7C

WORK SHEET
WESTINGHOUSE FORM 8295 E

32 HZ. DATA

OUTPUT V_{AC} PEAK	OUTPUT I_{AC} PEAK	INPUT V_{DC}	INPUT I_{DC} AMPS	Wx at "FABE"	MOTOR RPM	DYNAMOMETER FLD. EXCIT %	
100	50	124.4	9.6	2980	634	2	LIGHT LOAD
80	45	124.4	8.6	1B38	631	2	
70	45	124.4	8.6	14E0	628	2	
60	42	124.4	8.2	0FE0	622	2	
50	37	124.4	7.8	0C53	612	2	
40	43	124.4	7.8	0B40	580	2	
35	47	124.4	8.2	0CB6	553	2	
30	62	124.4	8.2	17B0	299	2	
100	67	122.4	27.0	3290	617	6	MEDIUM LOAD
80	67	122.3	26.6	27A0	603	6	
70	72	122.3	27.0	26BB	583	6	
65	75	122.2	28.4	2880	570	6	
60	87	122.1	28.6	3020	532	6	
105	95	119.8	55.8	50C2		10	HEAVY LOAD
100	95	119.9	55.8	4F10	591	10	
95	100	119.8	56.8	5126		10	

FIGURE 7D

WORK SHEET
WESTINGHOUSE FORM 8295 E

OUTPUT		INPUT		Wx at "FABE"		MOTOR	DYNAMOMETER FLD. EXCIT.	45 HZ DATA
VAC PEAK	IAC PEAK	VDC	IDC AMPS	HEX	DEC	RPM	%	
100	45	123.8	13.8	26AE		886	2	
80	40	123.8	13.2	1A80		877	2	
70	40	123.8	13.0	16D0		869	2	
60	42	123.8	13.0	1520		851	2	LIGHT LOAD *
55	47	123.7	13.2	1630		833	2	
50	55	123.7	13.4	1A40		798	2	
								60 HZ DATA
100	45	123.0	19.6	2380		1174	2	
90	45	123.0	19.4	1FD0		1166	2	
80	47	123.0	19.4	1E6B		1154	2	LIGHT LOAD *
70	55	123.0	19.8	2080		1129	2	
60	75			4800		833	2	

* INSUFFICIENT VOLTAGE AVAILABLE FOR FULLY LOADING THE MOTOR AT FREQUENCIES > 32 HZ.

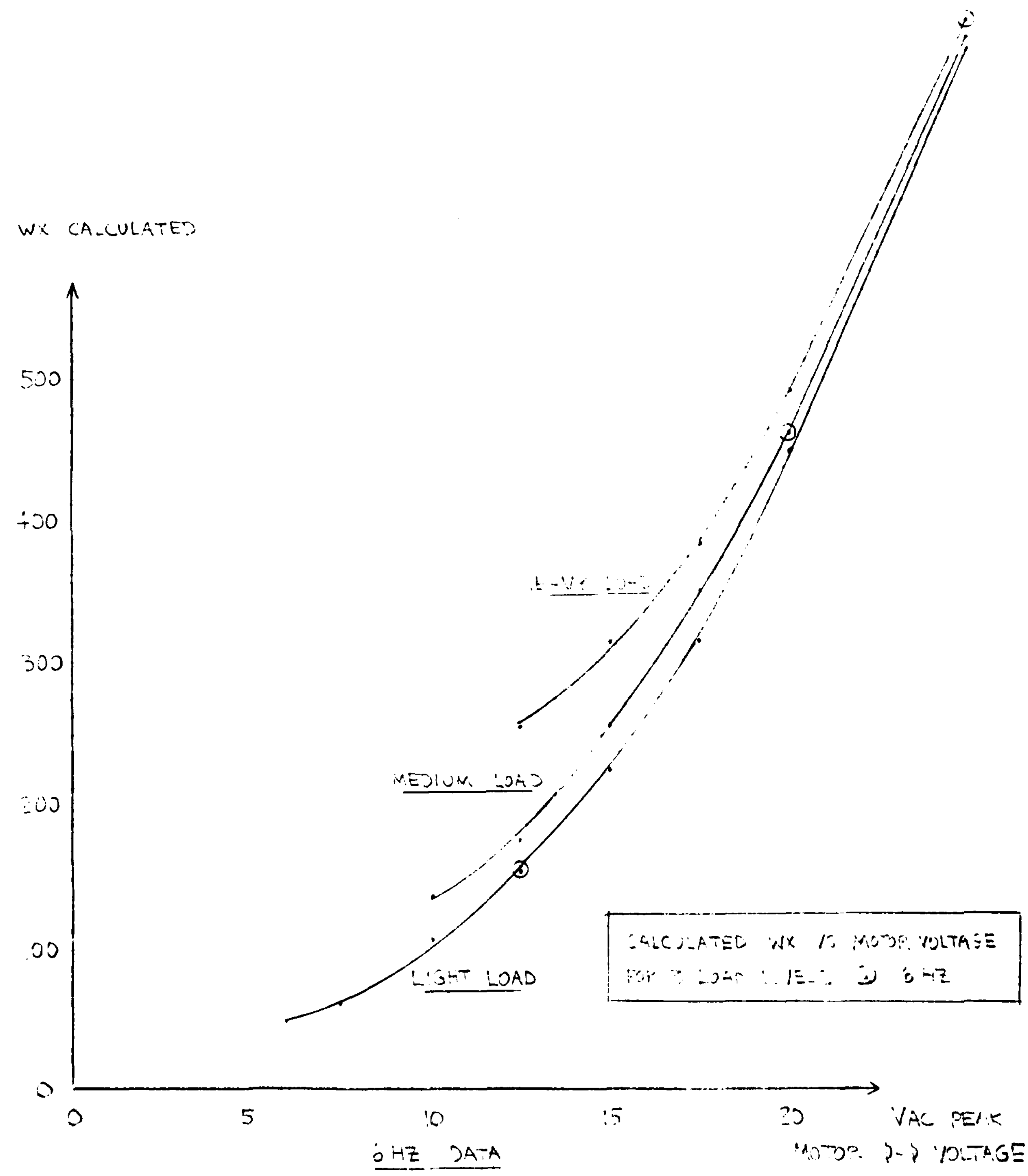
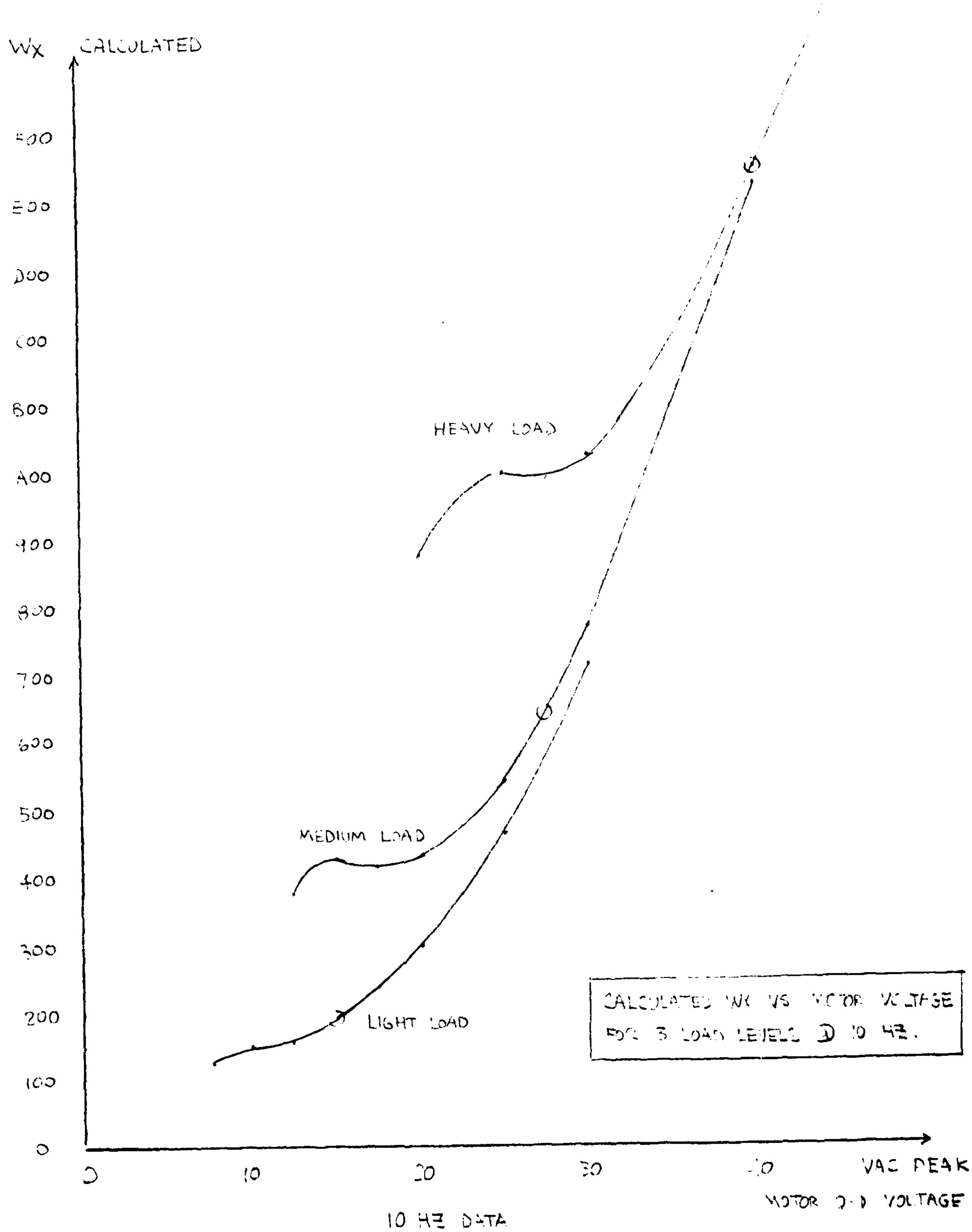
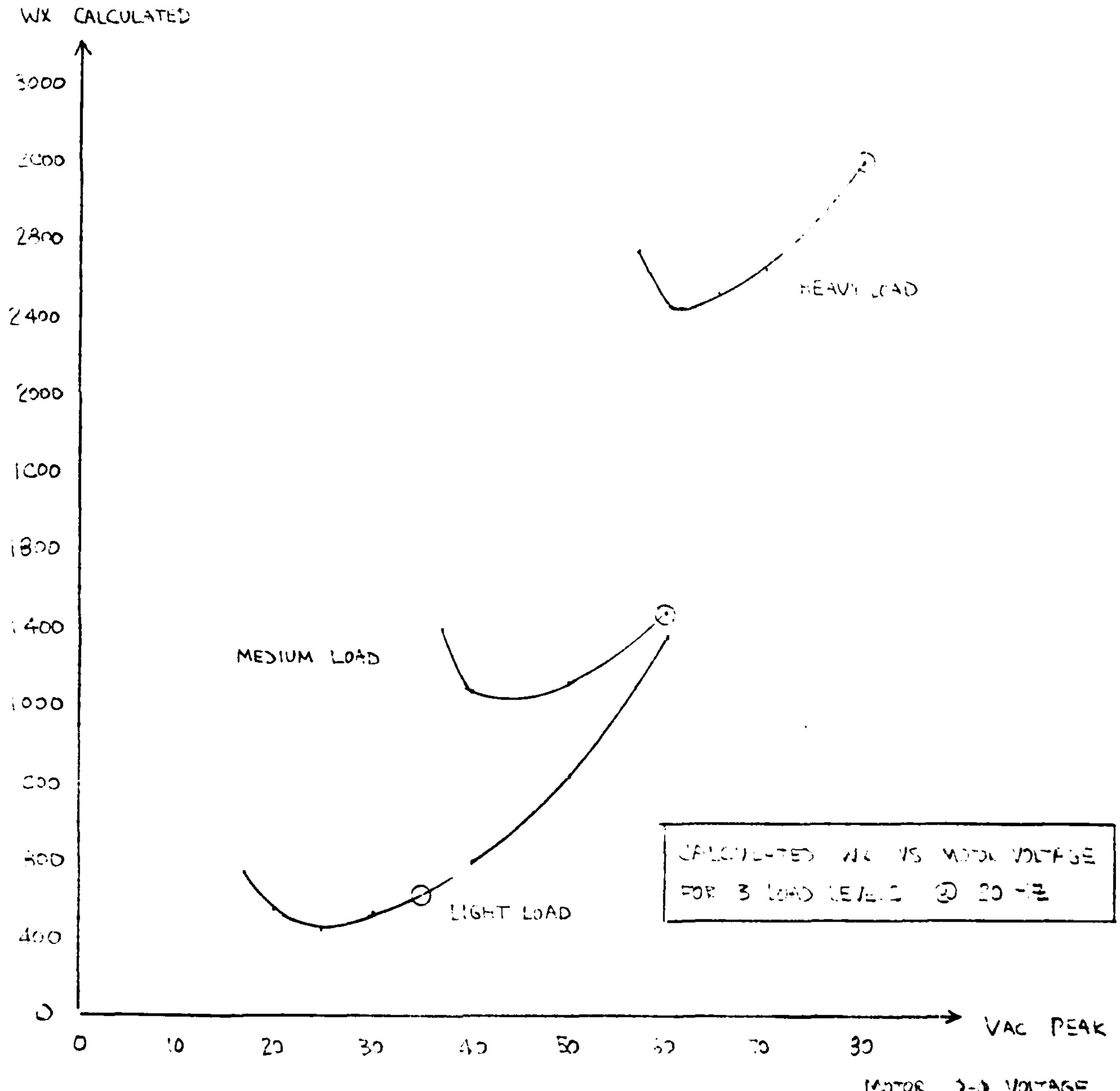

WX CALCULATED
500
400
300
200
100
0
0
5
10
15
20
VAC PEAK
MEDIUM LOAD
LIGHT LOAD
6 HZ DATA

FIGURE 8



Wx CALCULATED
HEAVY LOAD
MEDIUM LOAD
LIGHT LOAD
CALCULATED WX VS MOTOR VOLTAGE FOR 3 LOAD LEVELS @ 10 HZ.
VAC PEAK
MOTOR P-P VOLTAGE
10 HZ DATA

FIGURE 9



20 HZ DATA

FIGURE 10

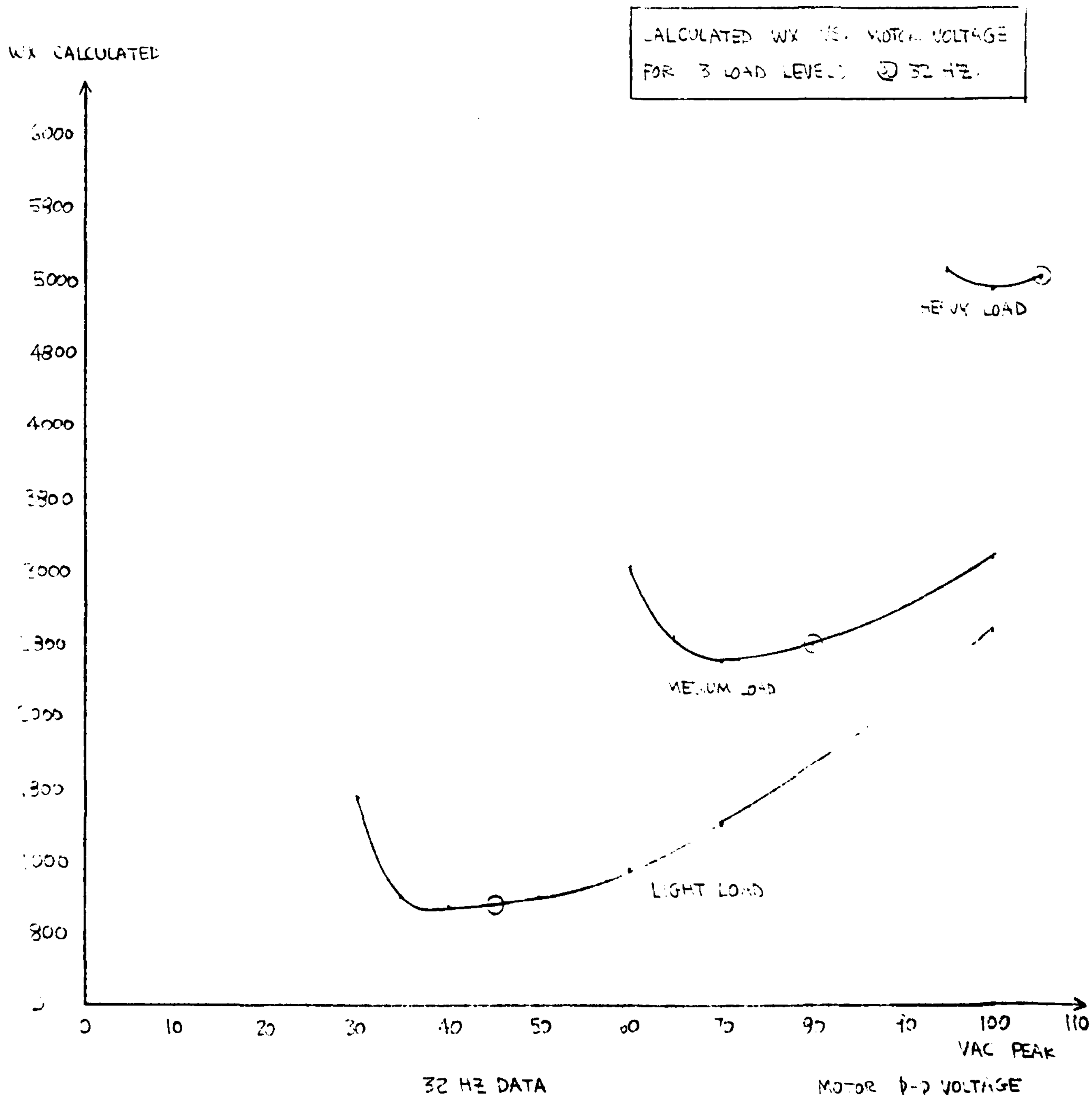
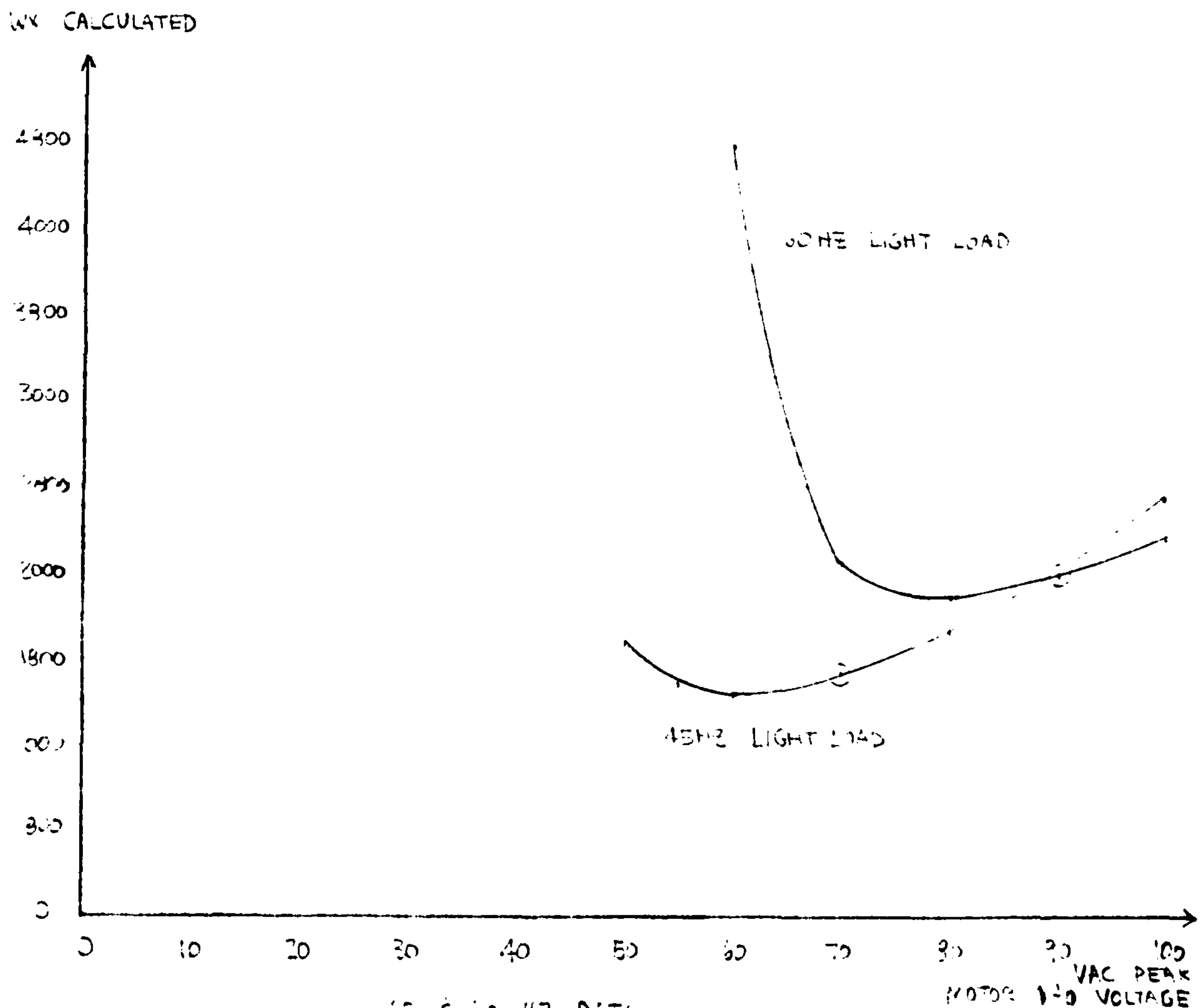

FOR 3 LOAD LEVELS @ 32 HZ
WX CALCULATED
5000
4800
4000
800
0
10
20
30
40
50
70
100
110
VAC PEAK
LOAD
LIGHT LOAD
32 HZ DATA
MOTOR P-P VOLTAGE

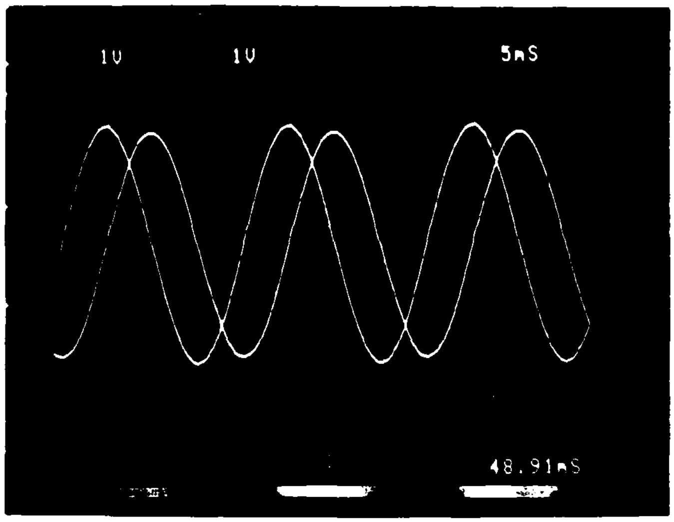
FIGURE 11

CALCULATED WX VS MOTOR VOLTAGE
FOR LIGHT LOAD @ 45HZ & 60 HZ



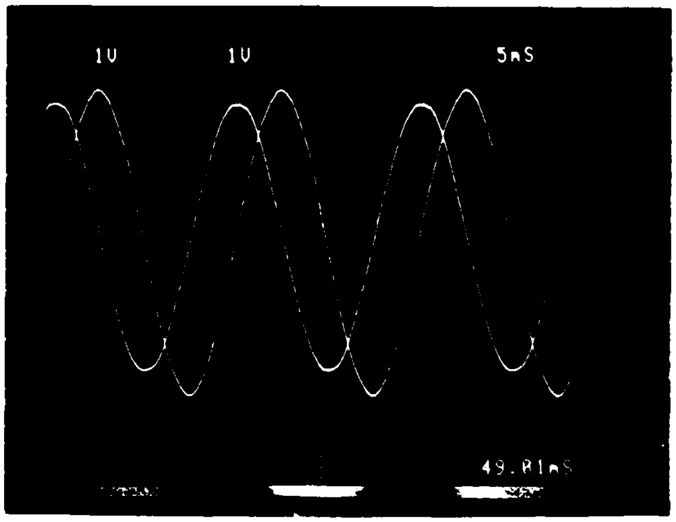
45 & 60 HZ DATA

FIGURE 12



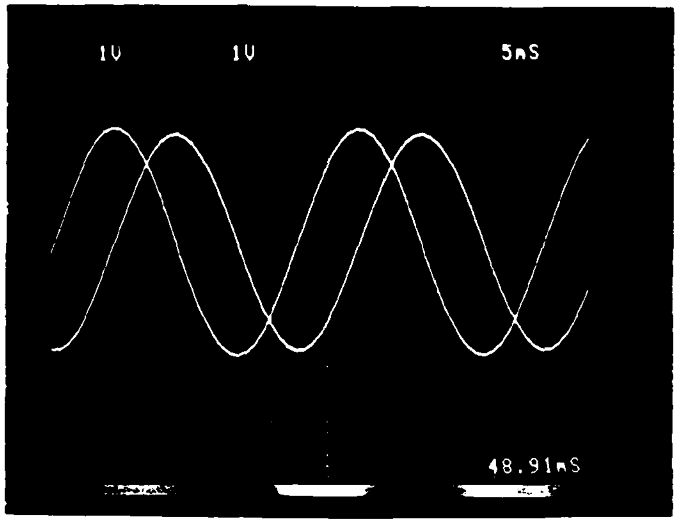
17

Vd, Vq at 60 Hz



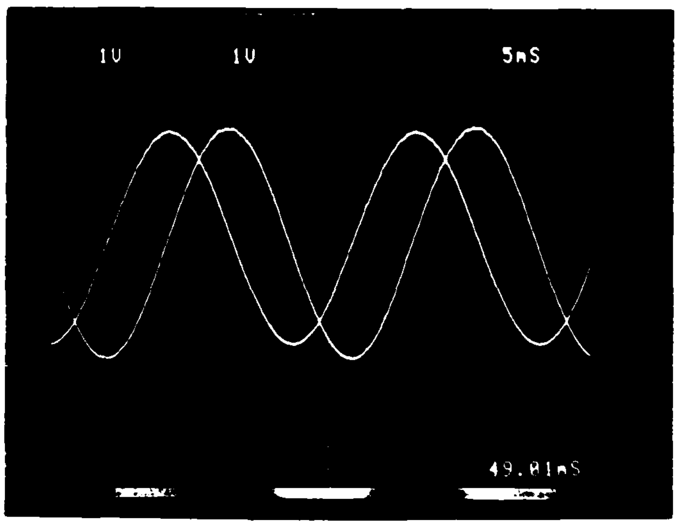
18

Id, Iq at 60 Hz



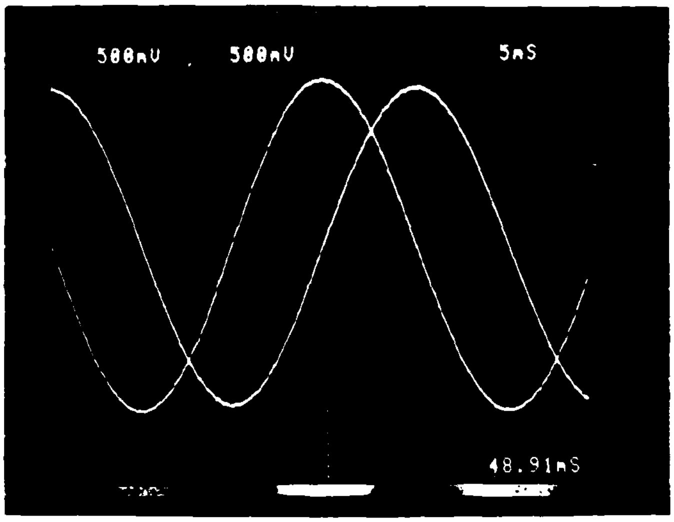
19

Vd, Vq at 45 Hz



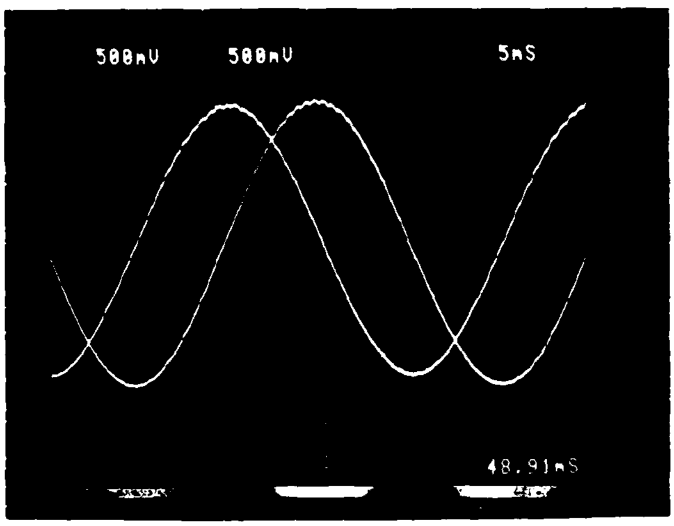
20

Id, Iq at 45 Hz



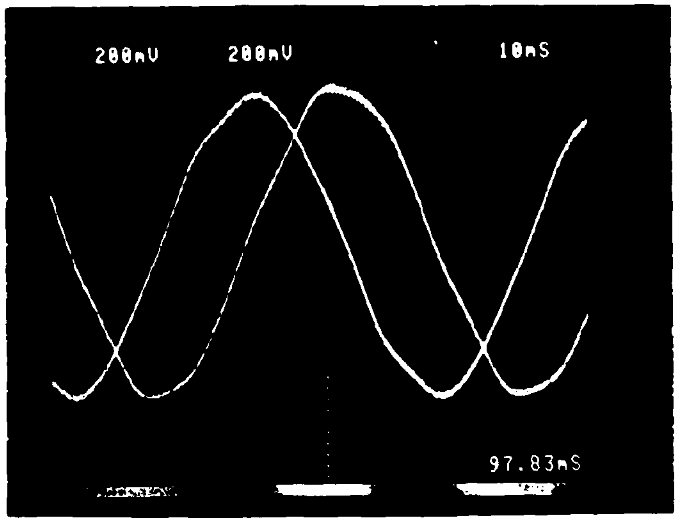
21

Vd, Vq at 30 Hz



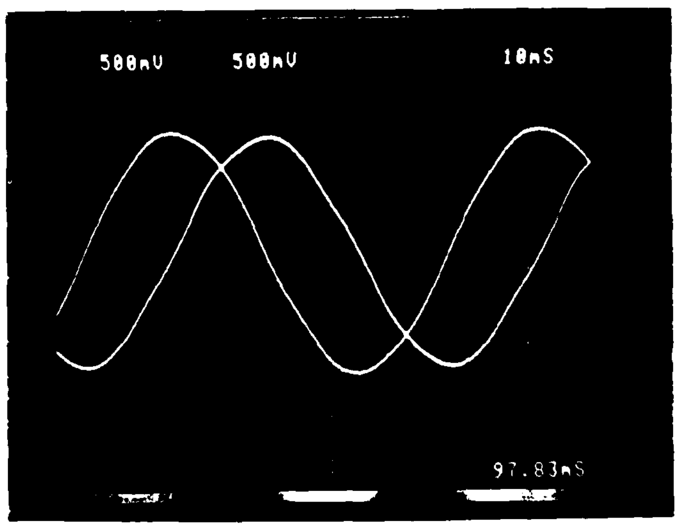
22

Id, Iq at 30 Hz



23

Vd, Vq at 15 Hz



24

Id, Iq at 15 Hz

90 AMPERES

PT-7511

SILICON NPN TRANSISTOR

MAXIMUM RATINGS	SYMBOL	PT-7511
Collector-Base Voltage	V_{CBO}	200V
Collector-Emitter Voltage	V_{CEO}	200V
Emitter-Base Voltage	V_{EBO}	10V
Peak Collector Current	I_{CM}*	90A
D.C. Collector Current	I_C	50A
Power Dissipation at 25°C Case Temperature	P_D	350W
Power Dissipation at 100°C Case Temperature	P_D	200W
Operating Junction Temperature Range	T_J	-65 to 200°C
Storage Temperature Range	T_A	-65 to 200°C
Thermal Resistance	θ_{JC}	0.5° C/W
Package		TO-63

ELECTRICAL CHARACTERISTICS (at 25°C unless noted)

TEST	SYMBOL	LIMITS PT-7511 MIN.	LIMITS PT-7511 MAX.	UNIT	TEST CONDITIONS
D.C. Current Gain*	h_{FE}	10	40		I_C=50A, V_{CE}=2V
D.C. Current Gain*	h_{FE}	5	–		I_C=90A, V_{CE}=4V
Collector Saturation Voltage*	$V_{CE(sat)}$	–	0.6	V	I_C=50A, I_B=5A
Collector Saturation Voltage*	$V_{CE(sat)}$	–	1.5	V	I_C=90A, I_B=18A
Base Emitter Voltage*	V_{BE}	–	1.5	V	I_C=50A, V_{CE}=2V
Base Emitter Voltage*	V_{BE}	–	2.5	V	I_C=90A, V_{CE}=4V
Collector-Emitter Breakdown Voltage*	$V_{CEO(sus)}$	200	–	V	I_C=200mA, I_B=0
Collector Cut-off Current	I_{CBO}	–	2.0	mA	V_{CB}=200V, I_{EB}=0
Collector Cut-off Current @ 150°C	I_{CBO}	–	10	mA	V_{CB}=100V, I_{EB}=0
Emitter Cut-off Current	I_{EBO}	–	1.0	mA	V_{EB}=8V, I_{CB}=0
Gain Bandwith Product Typ.	f_t	1.0	–	MHz	I_C=5A, V_{CE}=10V f=100KHz
Collector Capacitance	C_{obo}	–	1800	pf	V_{CB}=10V, f=1MHz
Switching Speed Typ.	t_r	–	2.5	µs.	
(PowerTech Test Circuit)	t_s	–	3	µs.	I_C=50A
	t_f	–	2.5	µs.	I_{B1}=10A, $-I_{B2}$=5A

*PW ≤ 300µs., D.C. ≤ 2%

![PowerTech logo]

PowerTech, Inc.

9 BAKER COURT · CLIFTON · N. J. 07011

TEST DATA

CUSTOMER Westinghouse P.O. # 876AQ[illegible] PRINT #

GROUP MT-1223 SUBGROUP

TEST CONDITIONS: Tc = 25°C LIMITS OR END POINTS:

UNIT #	@200mA BVceo	@8V Iebo	@200V [illegible]		@ 25CA Vce = 5V hFE	[illegible]		
	min 200V	max 15mA	max 10mA		min 100	max 3.0V		
001	254V	<100µA	<100µA		160	2.17		
002	252	<100	<100		175	2.07		
003	252	<100	<100		175	1.98		
004	248	<100	<100		167	2.03		
005	254	<100	<100		205	2.03		
006	248	<100	<100		154	2.14		
007	255	<100	<100		183	2.08		
007	255	250µA	<100		160	2.05		

SIGNED [illegible] DATE 1/[illegible]

A-Z

MT-1223

250 AMPERE POWERBLOCK POWER SYSTEM

MAXIMUM RATINGS	SYMBOL	MT-1223
Collector-Base Voltage	V_{CBO}	200V
Collector-Emitter Voltage	V_{CE}	200V
Emitter-Base Voltage	V_{EBO}	10V
Peak Collector Current	I_{CM}*	250A
D. C. Collector Current	I_C	250A
Power Dissipation at 25°C Case Temperature	P_D	975W
Power Dissipation at 100°C Case Temperature	P_D	600W
Operating Junction Temperature Range	T_J	-65 to 200°C
Storage Temperature Range	T_A	-65 to 150°C
Package:		PPS-500
Thermal Resistance	θ_{JC}	0.17°C/W

ELECTRICAL SPECIFICATIONS (at 25°C unless otherwise noted)

TEST	SYMBOL	MIN.	MAX.	UNITS	TEST CONDITIONS
D. C. Current Gain*	h_{FE}				I_C= V_{CE}=
D. C. Current Gain*	h_{FE}	100			I_C=250A, V_{CE}=5V
Collector Saturation Voltage*	$V_{CE(sat)}$		2.5	V	I_C=250A, I_B= 5A
Collector Saturation Voltage*	$V_{CE(sat)}$			V	I_C= I_B=
Base Emitter Voltage*	V_{BE}		-	V	I_C= V_{CE}=
Base Emitter Voltage*	V_{BE}		3.0	V	I_C=250A, V_{CE}=4V
Collector-Emitter Breakdown Voltage* †	$V_{CE(sus)}$	200		V	I_C=50mA
Collector Cutoff Current**	I_{CES}		10	mA	V_{CB}= 200V, R_{BE}=0
Emitter Cutoff Current***	I_{EBO}		15	mA	V_{EB}= 8V, I_C=0

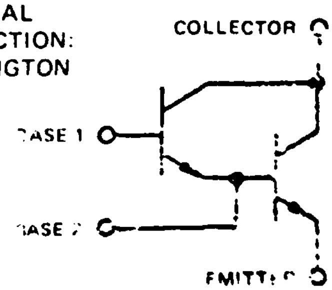
* ≤ 300 μsec. DC ≤ 2%

** Base #1 connected to Base #2

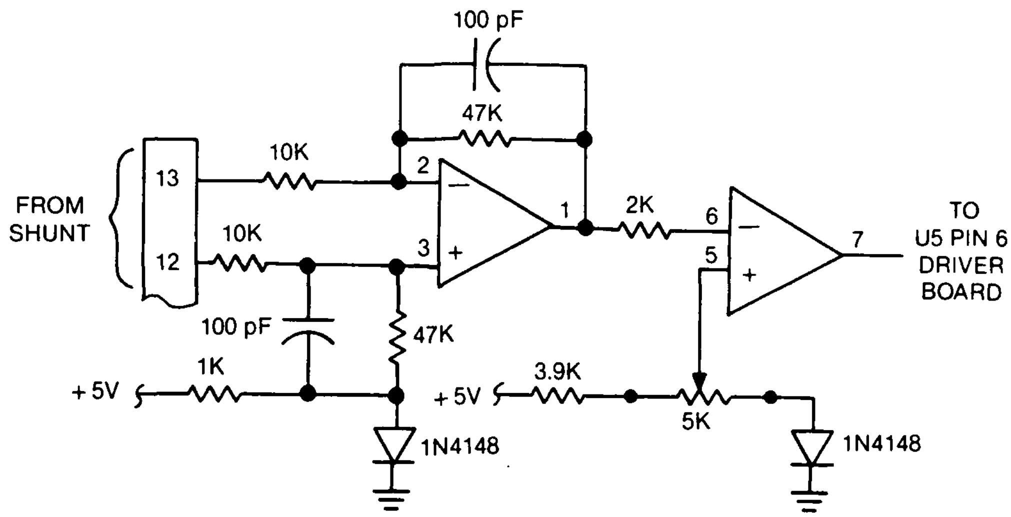
*** Base #2 open circuit

† $R_{B_1B_2}$ = 100 ohms, R_{B_2E} = 10 ohms

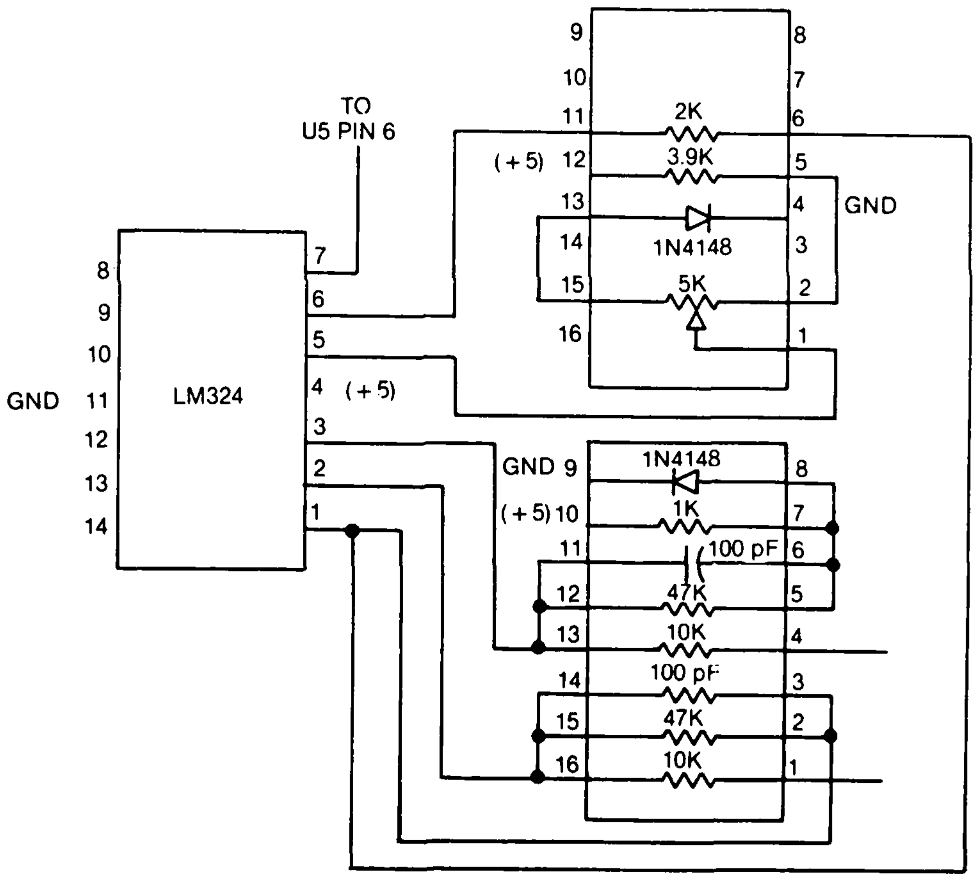
INTERNAL CONNECTION:
DARLINGTON



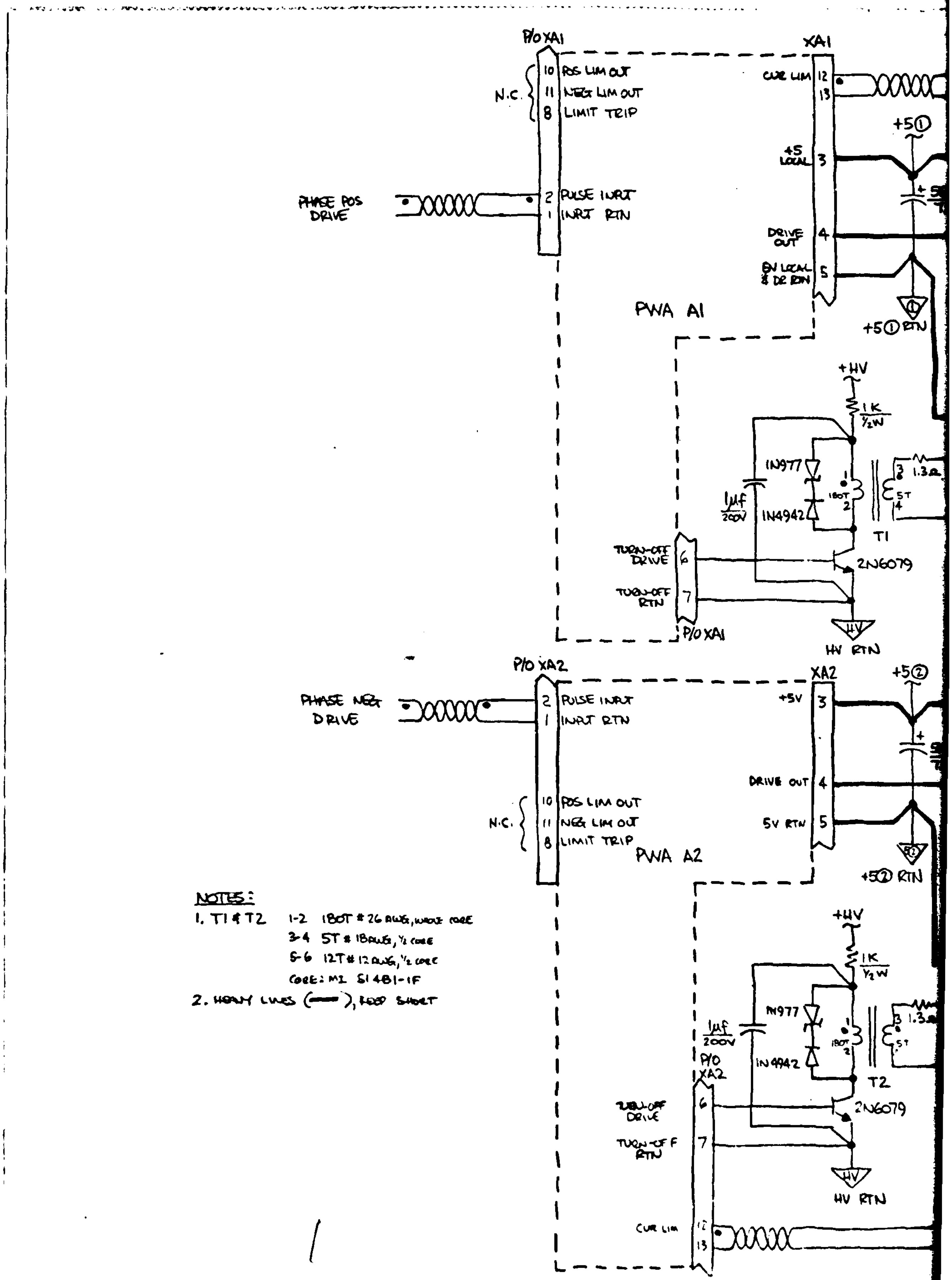
APPENDIX B



100 pF
47K
10K
FROM SHUNT
13
12
10K
2
3
1
2K
6
5
7
TO U5 PIN 6 DRIVER BOARD
100 pF
47K
1K
+5V
3.9K
+5V
5K
1N4148
1N4148

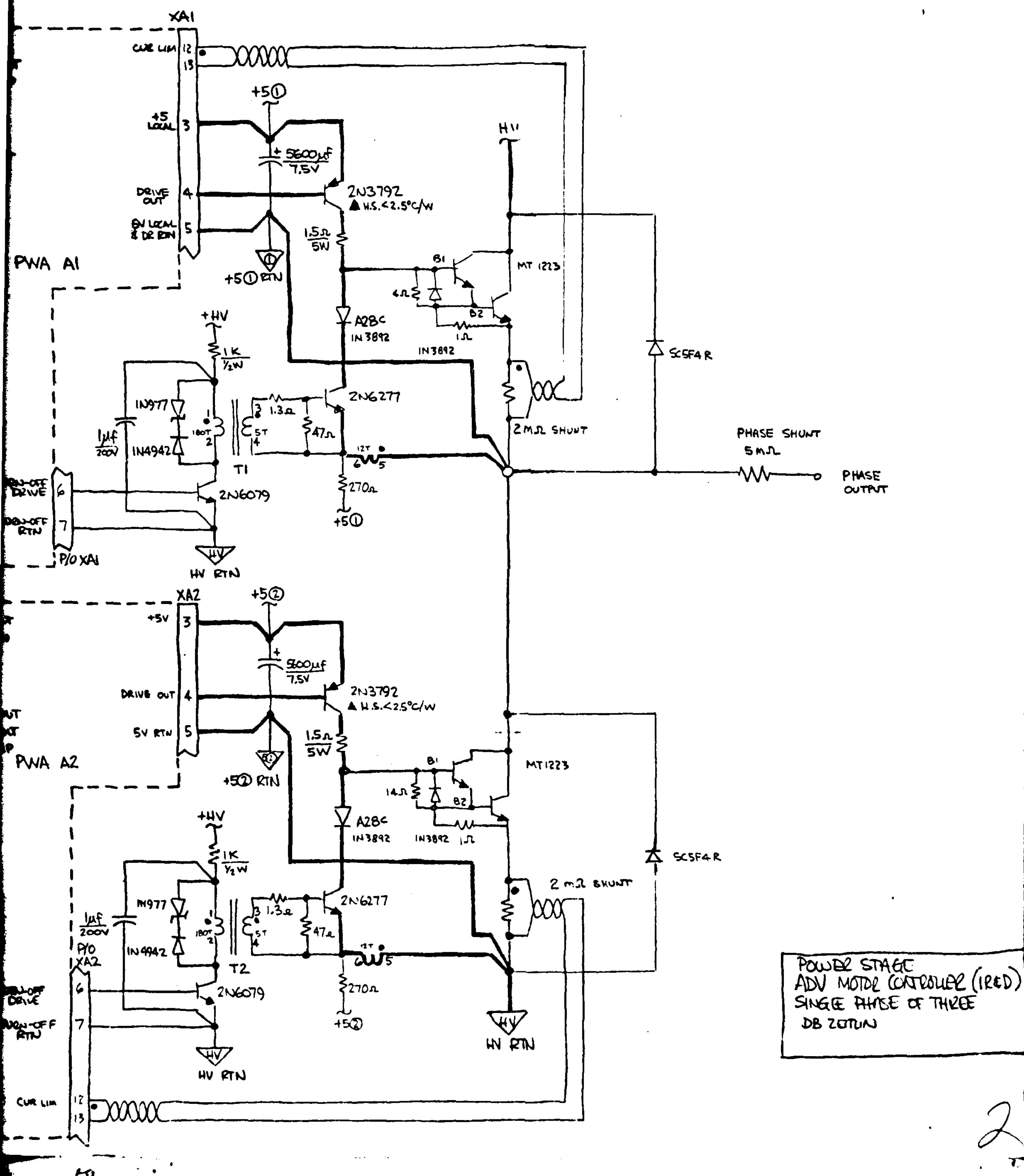


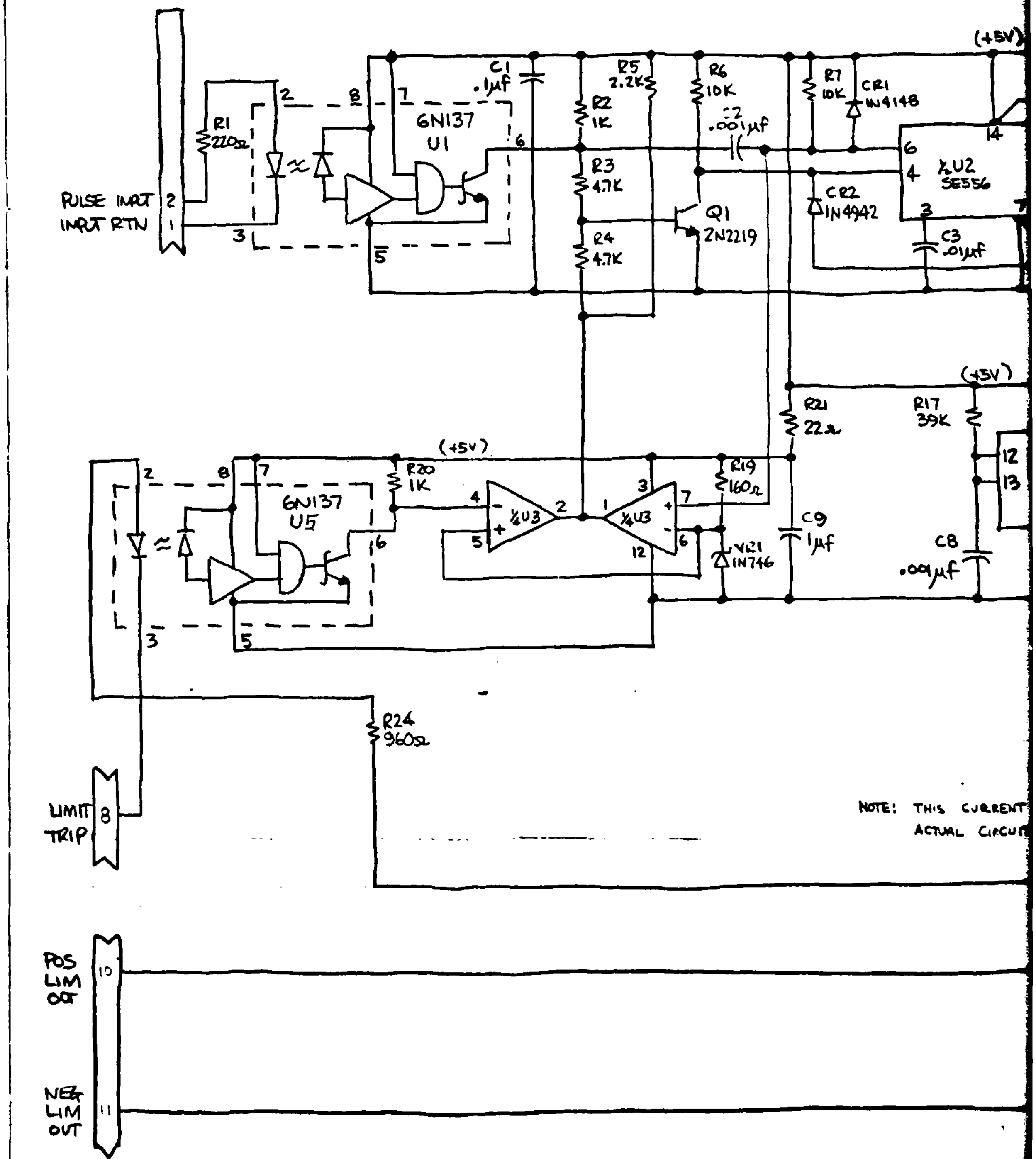
TO U5 PIN 6
LM324
GND
2K
(+5)
3.9K
GND
1N4148
5K
(+5)
GND 9
1N4148
(+5) 10
1K
100 pF
47K
10K
100 pF
47K
10K



NOTES:

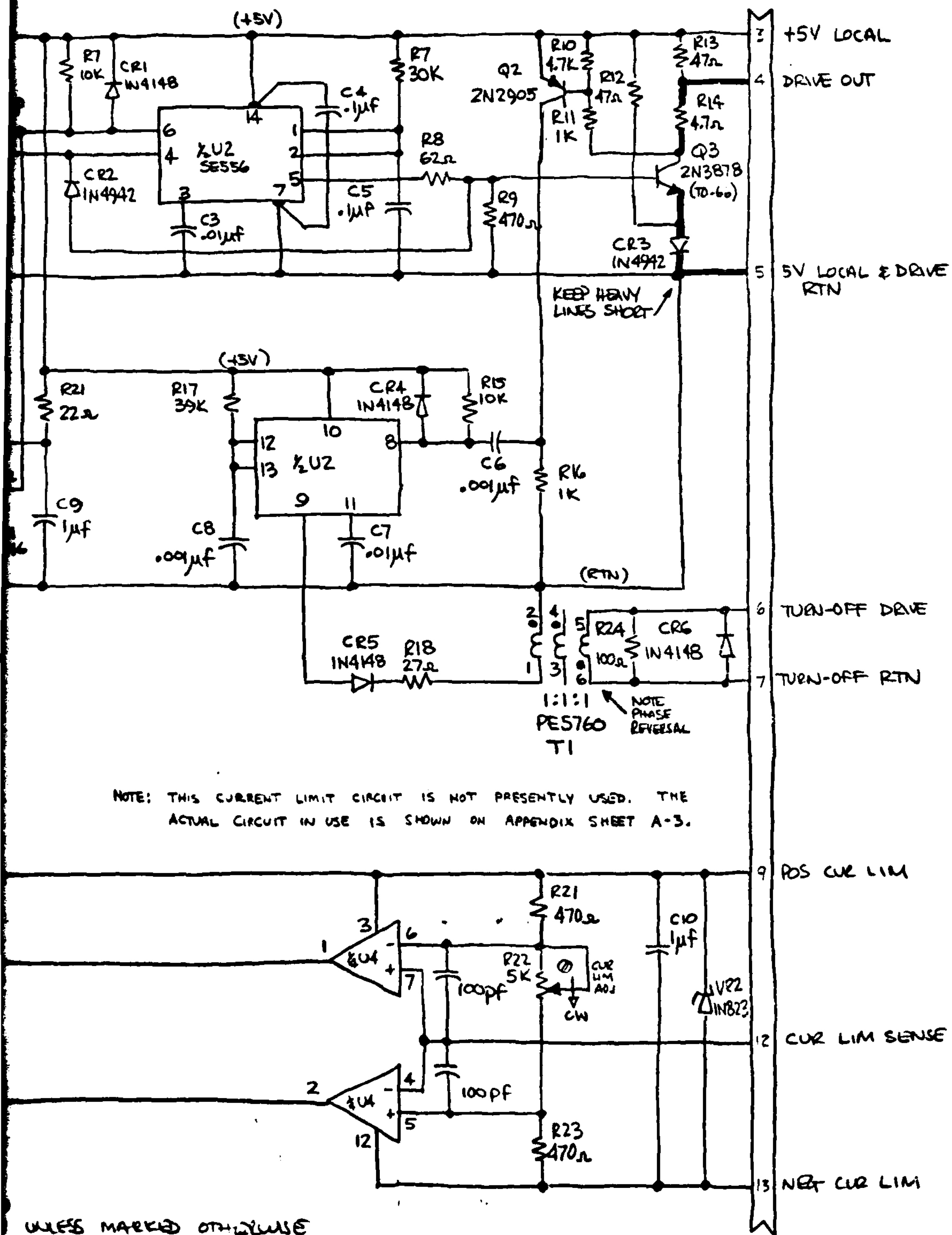
1. T1 & T2   1-2 180T #26 AWG, whole core
    3-4 5T # 18AWG, ½ core
    5-6 12T # 12AWG, ½ core
    CORE: MI 51481-1F
2. HEAVY LINES (▬▬), KEEP SHORT


XA1
CUR LIM
+5 LOCAL
DRIVE OUT
0V LOCAL & DR RTN
PWA A1
+5① RTN
+5①
5600µf 7.5V
2N3792
H.S.<2.5°C/W
1.5Ω 5W
HV
MT 1223
B1
B2
A28C
IN3892
1Ω
SC5F4R
+HV
1K ½W
IN977
IN4942
1µf 200V
2N6277
1.3Ω
47Ω
T1
2N6079
270Ω
2mΩ SHUNT
PHASE SHUNT 5mΩ
PHASE OUTPUT
ON-OFF DRIVE
ON-OFF RTN
P/O XA1
HV RTN
XA2
+5②
+5V
DRIVE OUT
5V RTN
PWA A2
+5② RTN
MT1223
14Ω
2 mΩ SHUNT
T2
P/O XA2
CUR LIM
POWER STAGE
ADV MOTOR CONTROLLER (IR&D)
SINGLE PHASE OF THREE
DB ZOTTAN
2



NOTES:

1. 13 CONN. PINS USED
2. ALL RESISTORS RC07 UNLESS MARKED OTHERWI
3. ALL CAPACITORS CK06 UNLESS MARKED OTH
4. Q3 IS TO-66 STYLE
5. U1 & U5 ARE 8 PIN DIP - 6N137
6. U2 IS 14 PIN DIP - SE556
7. U3 & U4 ARE 14 PIN DIP - LM139
8. CUR LIM IS SET UP FOR 25A MINIMUM ADJUSTMENT (4TO



UNLESS MARKED OTHERWISE
KO6 UNLESS MARKED OTHERWISE
NLE
N DIP - 6N137
- SE556
N DIP - LM139
2 25A MINIMUM ADJUSTMENT (470Ω-5K-470Ω)

IR&D MOTOR CONTROLLER
DRIVER BOARD

DB ZEITLIN

2

DAT

FILM

6